Word Meaning Representation in Neural Language Models : Lexical Polysemy and Semantic Relationships

Aina Garí Soler

université
PARIS-SACLAY

Word Meaning Representation in Neural Language Models: Lexical Polysemy and Semantic Relationships

Représentation du Sens des Mots dans les Modèles de Langue Neuronaux : Polysémie Lexicale et Relations Sémantiques

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 580, Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat: Informatique
Unité de recherche: Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique, 91400, Orsay, France
Référent: Faculté des sciences d'Orsay

Thèse présentée et soutenue à Paris-Saclay, le 24/06/2021, par

Aina GARÍ SOLER

Composition du jury

Pierre ZWEIGENBAUM Directeur de Recherche, CNRS (LISN)	Président
Eneko AGIRRE Professeur, Euskal Herriko Unibertsitatea	Rapporteur & Examinateur
Chloé CLAVEL Maître de Conférences-HDR, LTCI, Télécom-Paris, Institut Polytechnique de Paris	Rapportrice & Examinatrice
Malvina NISSIM Maître de Conférences, University of Groningen	Examinatrice

Direction de la thèse

Alexandre ALLAUZEN Professeur, Université Paris-Dauphine, ESPCI	Directeur de thèse
Marianna APIDIANAKI Chargée de Recherche, CNRS & Chercheuse, Université d'Helsinki	Co-encadrante

Thèse de doctorat

NNT: 2021UPASG043

Acknowledgements

Pursuing a PhD may seem like a very solitary endeavour, and this is in part true. It involves a great deal of individual work, but there are many people who have – consciously or not – contributed to this thesis in countless different ways.

First and foremost, I thank my supervisor, Marianna, without whom this document, and all the work behind it, would not have been possible. I thank her for her great involvement in this thesis and her creative ideas; for virtually always being available and ready to offer guidance, despite the distance. Especially, for always looking for what's best for me, even before I started my PhD, and for putting up with my stubbornness and my mistakes along the way. I also thank Alexandre, for his help especially in the initial stages of the thesis, and for always being ready to help when I needed it.

I am very grateful to the members of the jury: Eneko Agirre, Chloé Clavel, Malvina Nissim and Pierre Zweigenbaum. It's an honor for me that they accepted to read and evaluate this work. I thank them for the effort in reading the manuscript and their very interesting questions during the defence. I also want to thank Sebastian Padó (and Pierre, again) for being part of my mid-term defence committee and for their valuable feedback.

I also want to thank Jose Camacho Collados and Anne Cocos, for their help in some of the experiments described in the thesis.

Looking at this document, I can't help but remember three wonderful women, my past supervisors: Lonneke, Mariona and Toni. They helped me be where I am today and I am grateful for everything they taught me.

I send a big thank you to Sophie, Laurence and Cristelle for their efficient administrative support; to Olivier, William, Eric, Jean-Claude and Laurent for keeping my machines running and my models training; and to Elisabeth and Nicolas for their technical help for my defence. Also to Anne Vilnat, for her help with matters related to the doctoral school.

Academic, technical and administrative help have been crucial, but not less important than the nice atmosphere in the working environment. I have been lucky to be in a lab (the LIMSI, now LISN) with many fellow PhD students, interns, postdocs and permanents, with whom I've had great moments in and outside the lab. I have really enjoyed our time together and how we have supported each other: rehearsing presentations, giving feedback, sharing ideas, frustrations, experience and advice. A special mention to Aman and José for their help and patience every time I knocked their door with technical questions; to Zheng for our fascinating discussions; to Leonardo for his support, advice, and kindness; to Yuming and Sam for their

kind ear and for so much more. I have a soft spot for my lovely office mates, however short our time together was: Franck, Djidji, Sharleyne and Shu. And with their permission, I thank the longest-lasting, best office mate I could have ever asked for, Syrielle. I thank her for lighting up the most stressful day with her smile, positivity, and contagious enthusiasm; as well as for her help with proof-reading. I also thank Hussein, for being there since literally day one, for sharing his –often very much needed– chill perspective on things. Also Cristian, for spicing up afternoons with his ability to add humour to every situation. And to everyone else, thanks for the moments shared together: Alban, Álvaro, Benjamin, François, Hicham, Hugues, Jitao, Khoa, Lauriane, Léo, Marc, Matthieu, Minh Quang, Paul, Rachel, Robin, Sofiya, Soyoung...

I wish I could have spent more time with all of them, especially with newcomers, but you know what happened in 2020. As much as I resent Covid for multiple reasons (like snatching my deserved opportunity to travel to Punta Cana and Mexico, and taking away lab life), I must say that writing a thesis is much easier when you know you're not "missing out" on much. Still, I really appreciate the opportunity that this PhD gave me to participate in conferences and a summer school, either abroad or on-line, which were tremendously enriching experiences and where I met very interesting people, with whom I had enjoyable discussions – among others, Flora and Laura.

Moral support has been key too. I am very fortunate to have so many lovely people around willing to lend an ear and share my joy and sorrows. I thank my "local" friends Fanny, Hyunwoo, Kristina, Luis, Sasha and Thomas. I'm also grateful to so many "old" friends who are far away: Aitor, Duncan and Gisela, to whom I send my best wishes for their PhDs; Adrià, Francesc, Juanca, Naiara, Sergi, Sònia and my beloved school friends, especially Marina and Marta.

Finally, I am infinitely grateful to my family, especially my parents and Alex. For everything they've done during, but also before, my PhD. For their understanding, their love, for always being there despite the fact that I could not always be there in return. I thank Alex for his daily patience, help and unconditional support, and for brightening up the whole journey.

I have learnt so much from every one of them. In a way, they have shaped who I am today: if I were a word, they would be my context.

... Although not really. People are not like words. Words have synonyms; you are all irreplaceable.

Abstract

Word embedding representations generated by neural language models encode rich information about language and the world. In this thesis, we investigate the knowledge about word meaning encoded in embedding representations and propose methods to automatically enhance their quality. Our main focus is on contextual models which generate representations that capture the meaning of word usages in new contexts. These models have dominated the NLP and Computational Linguistics fields and open exciting new possibilities for lexical semantics research.

The central axis of our research is the exploration of the knowledge about lexical polysemy encoded in word embedding models. We access this knowledge through usage similarity experiments and automatic substitute annotations assigned by the models to words in context. We study the representations produced by the models in their raw form, and explore the impact that their enrichment with external semantic knowledge has on their quality. We evaluate the representations intrinsically on the tasks of usage similarity estimation, word sense clusterability and polysemy level prediction. Additionally, we employ contextualised representations for detecting words' semantic relationships, specifically addressing the relative intensity of scalar adjectives. Adopting an interpretation stance, we investigate the knowledge that the models encode about noun properties as expressed in their adjectival modifiers, and the entailment properties of adjective-noun constructions.

Our experiments involve a wide range of contextualised models which we compare to models that produce static word representations. The majority of our analyses address English but we also test our assumptions and methodology in a multilingual setting which involves monolingual and multilingual models in other languages. Our results demonstrate that contextualised representations encode rich knowledge about word meaning and semantic relationships acquired during model training and further enriched with information from new contexts of use. We also find that the constructed semantic space encodes abstract semantic notions, such as the notion of adjective intensity, which can be useful for intrinsic lexical semantic analysis and in downstream applications. Our proposed methodology can be useful for exploring other intrinsic semantic properties of words and their semantic relationships in different languages, leading to a better understanding of the knowledge about language encoded in neural language models.

Résumé

Les modèles de langue neuronaux sont entraînés sur de vastes quantités de données et génèrent des plongements lexicaux encodant des informations riches sur la langue et le monde. Dans cette thèse, nous étudions les connaissances sémantiques encodées dans ces plongements et proposons des méthodes automatiques pour en améliorer la qualité. Nous nous concentrons principalement sur des modèles contextuels récents qui génèrent des représentations décrivant le sens de mots en contexte. Nous comparons ces représentations à celles générées par des modèles de plongement antérieurs, qui ne sont pas contextualisées et qui se situent au niveau des mots. Les modèles contextuels se sont imposés dans les domaines du Traitement Automatique des Langues (TAL) et de la linguistique computationnelle, et ouvrent de nouvelles possibilités extrêmement intéressantes pour la recherche en sémantique lexicale.

L'axe central de notre recherche est l'exploration des connaissances sur la polysémie lexicale encodées dans les modèles de langue neuronaux. Nous accédons à ces connaissances par le biais d'expériences qui mesurent la similarité entre usages de mots, et en s'appuyant sur des annotations de substituts automatiquement attribuées par les modèles à des occurrences de mots en contexte. Ces annotations décrivent le sens des différentes occurrences et reflètent leur similarité sémantique. Nous étudions les représentations produites par les modèles sous leur forme brute et explorons, dans un cadre de « fine-tuning », l'impact de leur enrichissement avec des connaissances sémantiques externes sur leur qualité. Nous évaluons les représentations intrinsèquement sur les tâches d'estimation de la similarité d'usages, de prédiction de la facilité de partitionnement de l'espace sémantique des mots dans des sens différents, et de prédiction de leur niveau de polysémie. De plus, nous utilisons des représentations contextualisées pour détecter des relations sémantiques entre les mots, plus spécifiquement en abordant l'intensité relative des adjectifs scalaires. Dans une perspective d'interprétation, et en utilisant des questions de type Cloze, nous étudions les connaissances que les modèles encodent sur les propriétés des substantifs telles qu'elles sont exprimées dans leurs modifieurs adjectivaux, ainsi que les propriétés d'implication caractérisant les constructions adjectif-substantif.

Nos expériences explorent un large éventail de modèles contextualisés, comprenant ELMo et BERT, que nous comparons à des modèles qui génèrent des représentations statiques (non contextualisées) des mots, comme Word2Vec et GloVe. La majorité de nos analyses portent sur l'anglais mais nous testons également nos hypothèses et notre méthodologie dans d'autres langues (finlandais, français, espagnol et grec) en utilisant des modèles aussi bien monolingues que multilingues. Nous explorons aussi la localisation des connaissances sémantiques au sein

des modèles. Nos résultats démontrent que les représentations contextualisées encodent des connaissances riches sur le sens des mots et leurs relations sémantiques qui sont acquises lors de l'entraînement des modèles et qui sont, par la suite, enrichies par des informations provenant de nouveaux contextes d'utilisation. Nous constatons également que l'espace sémantique construit par ces modèles encode des notions sémantiques abstraites, comme la notion d'intensité des adjectifs, qui peuvent être utiles aussi bien pour l'analyse de la sémantique lexicale que dans des applications réelles. En outre, nos résultats mettent en évidence des différences entre les modèles monolingues et multilingues. Par rapport aux modèles de type BERT, précisément, nous observons qu'ils encodent des connaissances sémantiques moins précises dans des langues autres que l'anglais, et que la localisation de ces informations varie entre les différents modèles étudiés. La méthodologie proposée peut être utile pour explorer d'autres propriétés sémantiques intrinsèques des mots ainsi que leurs relations sémantiques dans différentes langues, conduisant à une meilleure compréhension des connaissances sur le langage encodées dans les modèles de langue neuronaux.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Neural Language Models (LMs) are able to generate vector representations of words that encode rich information about language and the world, which they learn from being exposed to large amounts of unannotated text. These models evolved from classical Vector Space Models (VSMs), where word representations were derived from co-occurrence matrices. Neural models rely on the same underlying principle as VSMs, the Distributional Hypothesis (Harris, 1954), which states that semantically similar words appear in similar contexts. However, instead of explicitly counting word co-occurrences, the models are trained to predict words in context (Baroni et al., 2014b). This results in word embeddings that reflect distributional similarity: words that occur in similar contexts have representations that are close to each other in the vector space. The first neural language models (Mikolov et al., 2013a) produced representations for word types. The limitation of this approach, known as the meaning conflation deficiency, is the inability to model the different senses of ambiguous or polysemous words, which are merged in a single vector. The only way to represent different senses of a word is through the combination of these embeddings (for example, combining the vector of *turn* with that of *fan* in "turn on the fan" to represent its VENTILATOR sense) (Erk and Padó, 2008). Multi-prototype and sense embeddings (Reisinger and Mooney, 2010; Iacobacci et al., 2015) overcome this limitation by proposing vectors corresponding to word senses. These approaches, however, are still limited in their capacity to represent meaning nuances that arise from contextual variation. Additionally, their integration into NLP models is not straightforward. During the course of this thesis, a new generation of deep contextual neural LMs emerged, including ELMo (Peters et al., 2018a) and BERT (Devlin et al., 2019). Relying on deep recurrent networks or attention mechanisms, these models generate embeddings for word usages in new contexts, which are referred to as contextualised representations. They have obtained state-of-the-art performance in numerous Natural Language Processing (NLP) tasks, and now constitute the predominant paradigm in the Computational Linguistics and NLP fields.

An important strand of work is focused on understanding what these new models actually learn about language and the world (Rogers et al., 2020). This thesis falls in this line of research,

and specifically explores different aspects of lexical semantics. Our goal is to understand what contextual models learn about the meaning of words. Capturing word meaning and the semantic relationships between words is crucial for language understanding, both for humans and machines. Lexical ambiguity is ubiquitous in language, and words can be related to each other in multiple ways. Knowing what meanings words can express, understanding their meaning when used in each context, and capturing relationships and similarities between words is important in virtually any application involving natural language.

The nature of contextualised embeddings, which represent word instances or tokens, opens up exciting possibilities and challenges in terms of methodology. The first question we address is: how well do these models represent word meaning in context? When a word is used in a sentence, its context helps determine the intended meaning. The contextualised representation of a word is precisely a function of the other words in the context, and as such it has some kind information from the context. In this thesis, we explore whether this contextual information allows models to determine words' meaning. We do so by evaluating the models' ability to identify meaning-preserving substitutes for words in context, and to determine the semantic proximity of word usages. Lexical substitutes can serve as a proxy for word meaning in context: in the sentence "My boss *fired* me", *fire* can be substituted by *sacked* but not by *shot*, which would conversely be a good substitute in "Soldiers *fired* at the enemy". Representations that properly model lexical meaning should be able to predict which substitutes are adequate in each case. Ideally, representations should also reflect the fact that the two usages of *fire* are very different from each other in terms of meaning. Importantly, these two tasks allow us to answer our question without the need of resorting to a sense inventory. Lists of senses are highly subjective and are defined by a number of non-linguistic factors (Kilgarriff, 1997). There is no unique way to establish boundaries between word senses, and different resources vary in the granularity of the senses proposed. For example, whether the usages of *cover* described by the sentences "*Cover* the meat with a lot of gravy" and "*Cover* the child with a blanket" are assigned the same or two different senses depends on the resource where we look them up.[1] Throughout this thesis, we avoid using lists of word senses for disambiguation, and instead evaluate models on semantic tasks where meaning is described in different ways.

Token-level contextualised representations also offer an exciting opportunity to investigate the semantic space made up of word instances. Thus, another question we want to answer is: do the semantic spaces built by contextual models reflect the ambiguity of words? Words can express one or multiple senses, which are more or less distinguishable from each other. For example, the two instances of *fire* described above are very different from each other, but the MUSICAL and COMPUTER senses of *keyboard* share some common traits. Through an analysis based on usage similarity estimations, we investigate how monosemous words and words at different polysemy levels are represented in the semantic space. When a word has multiple senses, we use the models to predict how easy it is to partition this semantic space into distinct senses (McCarthy et al., 2016).

[1]These examples come from the WordNet lexical database (Fellbaum, 1998) and illustrate two different senses of *cover* in this resource.

Apart from investigating the knowledge that contextualised representations encode about individual words, we also study how they capture semantic relationships between words. We focus on two specific relationships: the relative intensity of scalar adjectives and the relation between nouns and adjectives describing their properties. Scalar adjectives may have similar meaning but differ in intensity (e.g. *good* and *fantastic*). Modelling this relation is important, especially because of its entailment properties: models should be able to tell that a *fantastic restaurant* is *good*, but a *good restaurant* is not necessarily *fantastic*. It can also serve to determine the subjectivity of a text, and can help language learners to distinguish between near-synonyms. The relationship between nouns and adjectives in adjective-noun (AN) constructions also has interesting entailment properties. Adjectives describing prototypical properties of a noun do not add new information, and hence entailment between the noun (N) and the AN holds bidirectionally in these cases (a *strawberry* and a *red strawberry* denote the same concept). This is not the case with most adjectives, however, which often restrain the scope of the noun to a subset of the entities it denotes (e.g. *white rabbit*).

Another important goal of our work is to improve the quality of the semantic information in contextualised representations. Throughout the thesis, we explore different ways of enriching representations with external semantic knowledge, for example, using automatic substitute annotations. We evaluate the representations on specific tasks which reflect whether these strategies increase their sensitivity to lexical meaning. Furthermore, we propose methodology for exploiting the information encoded in the representations for performing specific tasks. For example, we present an efficient method for ranking scalar adjectives by intensity using contextualised representations.

Our experiments are centered on English, but we also test our assumptions using multilingual and monolingual models in French, Spanish, Greek and Finnish. Additionally, we compare contextualised to word type representations in our experiments to highlight the advantages of models that encode contextual information. We demonstrate that contextualised representations, especially the ones derived from the BERT model, encode rich knowledge about word meaning and semantic relationships acquired during model pre-training, which is combined with information from new contexts of use. The constructed semantic space reflects semantic properties of words (e.g. their polysemy), and encodes abstract semantic notions, such as adjective intensity. Our work, hence, leads to a better understanding of the knowledge learnt by neural language models about words and their meaning. Our methodology can be useful for exploring other semantic properties of words and enhancing the quality of contextualised representations from different models and in different languages.

1.2 Outline

We hereby provide a summary of each chapter of the thesis.

Chapter 2: Background and Related Work We start by introducing notions related to word meaning that are central to our work, and present ways of describing the meaning of a

word in context. We also introduce the main datasets that will be used in this thesis. After that, we present the Distributional Hypothesis of meaning, on which all word representation models we use are based. Our description of word vector representation approaches begins with traditional distributional models, and we then present more recent neural language models with a focus on those that generate contextualised representations. Finally, we review recent studies on the interpretability of contextual language models, which aim at understanding the kinds of information they contain. We describe common interpretability methodologies and recent findings about the semantic knowledge encoded in contextualised representations.

Chapter 3: In-context Lexical Substitution The lexical substitution task initially served as a means to evaluate word sense disambiguation (WSD) models without the need to resort to a pre-defined sense inventory as in traditional WSD settings. A model that understands the meaning of word instances should be able to predict, for example, that the instance of *fan* in the expression "turn on the *fan*" can be replaced with *ventilator* without a big shift in meaning but not with *admirer*, which is a synonym of a different sense of the word. In Chapter 3, we compare the performance of several context-sensitive models (such as context2vec (Melamud et al., 2016), ELMo (Peters et al., 2018a) and BERT (Devlin et al., 2019)) on the task of in-context lexical substitution. The task consists in ranking substitute candidates for a word according to how appropriate they are in a given context. We employ different methods proposed in past work to combine the representations, and propose an approach to enrich them using substitute-specific information. We find that BERT representations work better than others for lexical substitution.

Chapter 4: Word Usage Similarity Estimation Another way of testing the ability of the models to represent the contextualised meaning of words is through word usage similarity estimation. Ideally, we would expect the representations for these two instances of *fan*: "turn on the *fan*" and "the *fan* is not working" to be highly similar, and dissimilar from its representation in "I'm your biggest *fan*". We evaluate several contextual and sentence embedding models on this task, using similarity scores assigned by their representations. Additionally, we propose to combine contextualised word embedding similarities with automatic substitute annotations for better word usage similarity prediction. This approach relies on the idea that the substitute overlap between two word instances reflects their semantic similarity. For example, the first two instances of *fan* in the examples above can be substituted by *ventilator*, and this indicates their semantic proximity. Although substitute annotations help in this task, their quality is key for a good performance. We show that when substitute quality cannot be assured, BERT representations are on their own a good predictor of word usage similarity.

Chapter 5: Word Sense Clusterability Estimation This chapter focuses on word sense clusterability (McCarthy et al., 2016), a lexical semantic property that refers to the ease of partitioning a word into senses. Although the VENTILATOR and ADMIRER senses of *fan* are very distinct from each other, distinctions are not so clear for other words: the HUSBAND, SOLDIER

and HUMAN senses of *man*, for example, are all related. We would thus say that *fan* is easier to partition into senses and, consequently, more clusterable than *man*. Knowing the clusterability of a word can be useful in order to determine its optimal computational representation: a per-sense approach could be preferable for clusterable words, while contextualised representations might be more adequate for less clusterable words, where meaning is more subtly modulated by context variation. We extend McCarthy et al.'s (2016) approach for word sense clusterability estimation using contextualised representations and automatic substitute annotations, and experiment with new clusterability metrics. We also carry out a first attempt at scaling up clusterability prediction on a large corpus using BERT representations, and uncover BERT's sensitivity to collocational and contextual differences in the usage of words. Finally, we propose to modify BERT representations of clusterable words by turning them into multi-prototype representations, and investigate the impact of this modification on a word usage similarity task.

Chapter 6: Fine-tuning BERT for Lexical Meaning In this chapter, we focus on the BERT model. First, we perform a systematic exploration of how context variation that does not modify the meaning of a sentence nor that of its individual words affects representations. We do this by observing the changes in usage similarity across pairs of sentences that differ in a specific linguistic phenomenon. For example, in a model that accurately reflects words' meaning, we would expect the representations of *fan* in "I bought a *fan* yesterday" and "A *fan* was bought yesterday" to be highly similar. Then, we experiment with different ways of increasing BERT's sensitivity to lexical meaning. We do so by fine-tuning BERT models on different semantic tasks which involve deciding whether two word instances, or two sentences, have the same meaning. Results obtained in an in-context word similarity task show that our approach is beneficial for English models, even when the data for fine-tuning has been automatically created.

Chapter 7: Polysemy Level Prediction The word *fan* can express a lower number of senses than the noun *shot*, which can refer to the firing of a projectile, an injection or a small drink, among others. The monosemous word *hotel*, instead, only has one sense. Do pre-trained LMs encode information about the number of senses of a word, and, if this is the case, where does this knowledge come from? In Chapter 7 we answer these questions based on an exploration of words' semantic space in different languages. In our experiments, we use monolingual BERT models in English, French, Spanish and Greek and multilingual BERT. By using datasets with controlled sense distributions, we find that BERT representations –especially from the English model– reflect whether a word is monosemous or polysemous, and its degree of polysemy. This knowledge is present regardless of the contexts used to extract them, meaning it is acquired during pre-training. We additionally account for the correlation between word frequency and number of senses (Zipf, 1945) and for the relation of grammatical category and polysemy, by balancing the frequency and part of speech (PoS) distributions in our datasets.

Chapter 8: Scalar Adjective Identification and Ranking Scalar adjectives can have similar meanings, but express them at different degrees of intensity. For example, *interested* and *passionate* describe similar characteristics of a *fan*, but the latter is more intense. The difference in intensity between the two adjectives affects their entailment relation (*passionate* $\Rightarrow$ *interested*, but *interested* $\not\Rightarrow$ *passionate*). This notion of intensity, however, characterises specifically scalar adjectives. Relational adjectives, such as *electric* or *English*, serve to classify (McNally and Boleda, 2004) a noun and do not express intensity. In this chapter, we first explore the knowledge that BERT representations encode about the intensity of scalar adjectives. We propose a resource-lean method for scalar adjective ranking inspired from gender bias work (Bolukbasi et al., 2016) which involves comparing adjectives in a scale to a vector expressing intensity. Given the good performance of this method in English, we extend it to other languages. We translate existing datasets to French, Spanish and Greek to promote research on these languages. Finally, we build a dataset to evaluate BERT's capability to distinguish scalar from relational adjectives which do not contribute to the emotional tone of a text.

Chapter 9: Nouns' Semantic Properties and their Prototypicality In this chapter, we explore the knowledge that the BERT model encodes about noun properties and their prototypicality, as expressed in their adjectival modifiers. For example, when referring to the VENTILATOR sense of fan, we can say that nowadays most *fans* are *electric*, but only some of them are *metallic*. *Electric* and *metallic* are adjectives denoting properties of *fans* that differ in their prototypicality. We also investigate the entailment properties of adjective-noun (AN) constructions. Adjectives often restrict the reference scope of the noun they modify, leading to AN phrases where the forward entailment between AN and the head noun N holds (AN $\models$ N, e.g. *metallic fan* $\models$ *fan*), but backward entailment does not (N $\not\models$ AN, e.g. *fan* $\models$ *metallic fan*). However, when an adjective denotes a prototypical property of a noun, entailment holds in both directions (AN $\models$ N and N $\models$ AN, e.g. *electric fan* $\models$ *fan* and *fan* $\models$ *electric fan*). This is explained by the fact that these adjectives do not add new information about the noun, but rather emphasise one of its inherent properties. We carry out an extensive investigation of the knowledge the BERT model has of noun properties and their prevalence. Our findings suggest that BERT has marginal knowledge about the prototypicality of noun properties as reflected in the dataset used for evaluation, but it can learn to distinguish prototypical from other properties and predict entailment in supervised settings.

1.3 Publications related to this thesis

- Aina Garí Soler, Anne Cocos, Marianna Apidianaki and Chris Callison-Burch (2019). A Comparison of Context-sensitive Models for Lexical Substitution. In *Proceedings of the 13th International Conference on Computational Semantics* (IWCS 2019), 23-27 May, Gothenburg, Sweden. (Garí Soler et al., 2019c) **(Chapter 3)**.

- Aina Garí Soler, Marianna Apidianaki and Alexandre Allauzen (2019). Word Usage Similarity Estimation with Sentence Representations and Automatic Substitutes. In *Proceed-*

ings of the 8th Joint Conference on Lexical and Computational Semantics (STARSEM2019), Jun 6-7, Minneapolis, USA. (Garí Soler et al., 2019b) **(Chapter 4)**.

- Aina Garí Soler, Marianna Apidianaki and Alexandre Allauzen (2019). LIMSI-MultiSem at the IJCAI SemDeep-5 WiC Challenge: Context Representations for Word Usage Similarity Estimation. In *5th Workshop on Semantic Deep Learning* (SemDeep-5). (Garí Soler et al., 2019a) **(Chapter 4)**.

- Aina Garí Soler and Marianna Apidianaki (2020). MULTISEM at SemEval-2020 Task 3: Fine-tuning BERT for Lexical Meaning. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, Dec 12-13, Barcelona, Spain. (Garí Soler and Apidianaki, 2020b) **(Chapter 6)**.

- Aina Garí Soler and Marianna Apidianaki (2020). BERT Knows Punta Cana is not just beautiful, it's gorgeous: Ranking Scalar Adjectives with Contextualised Representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing* (EMNLP), Nov 16-20. (Garí Soler and Apidianaki, 2020a) **(Chapter 8)**.

- Aina Garí Soler and Marianna Apidianaki (2021). Scalar Adjective Identification and Multilingual Ranking. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1 (Long and Short Papers), Nov 6-11. (Garí Soler and Apidianaki, 2021b) **(Chapter 8)**.

- Aina Garí Soler and Marianna Apidianaki (2021). Let's Play `Mono-Poly`: BERT Can Reveal Words' Polysemy Level and Partitionability into Senses. To appear in *Transactions of the Association for Computational Linguistics* (TACL). (Garí Soler and Apidianaki, 2021a) **(Chapters 5 and 7).**

Chapter 2

Background and Related Work

In this chapter, we provide relevant theoretical background about word meaning and present different approaches to representing it computationally. We start by introducing several notions related to lexical ambiguity and discussing ways of describing the senses of a word. We also present the datasets and lexical databases used in this thesis (Section 2.1). We then describe several approaches to representing words with vectors, from traditional distributional models to current contextualised Transformer-based models (Section 2.2). Finally, we provide an overview of recent interpretability work, which aims at unraveling the knowledge contained in contextual language models and the representations derived from them (Section 2.3).

2.1 Lexical Ambiguity

2.1.1 Ambiguity, Polysemy and Vagueness Continuum

Words often have multiple senses. For example, *coach* can be used to refer to a *trainer*, but also to a *bus*. The interpretation of words may change from context to context: *soft voice* and *soft breeze* evoke different senses of the word *soft*. This variation is ubiquitous in human language (Cruse, 1986), and new usages of words keep naturally appearing through meaning extension mechanisms such as metaphor and metonymy. In fact, although most lemmas in the vocabulary are **monosemous** (i.e. they have only one sense), lemmas with multiple senses are used with higher frequency (Zipf, 1945).

These differences in word usage can be of a discrete, clear-cut nature, as in the *coach* examples above, which denote distinct referents. However, the differences can also be quite subtle, as with the word *thing*, whose interpretation varies with each context of use. The concrete meaning of the word may be left underspecified: for example, in "All the *things* she said", *thing* could refer to a speech, a joke, an apology, a confession, etc.

We say that *coach* is an **ambiguous** word, because its senses are unrelated to each other. *Thing*, instead, is a word with **vague** semantics because its interpretation varies subtly with every context of use. For these words, it is particularly difficult to establish a list of senses.

Ambiguity and vagueness are extremes in a continuum, in the middle of which we find polysemy (Tuggy, 1993) (Figure 2.1). **Polysemous** words have senses that are distinct, but

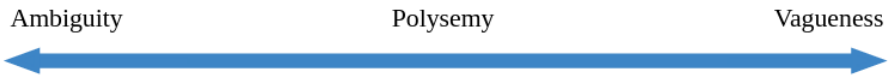


Figure 2.1: The continuum between ambiguity and vagueness, with polysemy in the middle.

have something in common with each other. One example are the usages of the word *soft* above. In this case, the context evokes different but related qualities: a *soft voice* is quiet and gentle, a *soft breeze* is also gentle, not strong. It is important to distinguish between polysemy and **homonymy**, which are two strongly related, and sometimes confused, phenomena. A word (or a lexeme) is said to be polysemous if it has multiple senses, and two words (or lexemes) are homonyms if they have the same form but different meaning. To distinguish a polysemous word from homonyms, linguists use etymological and sense-relatedness criteria – homonyms have different origins and their meanings are less related than those of a polysemous word (Lyons, 1995). Throughout this thesis, we simply use the term "polysemous" to refer to a word that has multiple meanings, regardless of their potential different origin. When those are highly distinct, we will refer to this word as ambiguous.

It is also worth noting that a word may have senses that are highly distinct from each other, and at the same time others that are closely related. For example, consider the verb *run*. The usages "I had to *run* to catch the bus" and "The script is *running*" describe completely different actions. The sense used in "I *ran* in a marathon" is very similar to that in the first sentence, but evokes a different way of moving one's feet: a controlled, stable pace vs. a rushed sprint. The last sentence can also be interpreted in the COMPETING sense of run.

2.1.2 Sense Enumeration and Delimitation

One way of accounting for differences in word meaning is proposing a **list of senses** for each word, as is traditionally done by lexicographers in dictionaries or lexical databases. The resulting meaning descriptions are useful as a reference for speakers of the language or language learners. One clear limitation of this approach, however, is its high subjectivity: whether the sense nuances of the word *soft* presented above are assigned two separate senses in a resource depends on the lexicographer, the intended audience or the purpose of the sense inventory that is being built (Kilgarriff, 1997). For polysemous and vague words, there is no unique correct way of establishing boundaries between senses. Different partitionings of words exist in different resources and are equally valid, despite varying greatly in terms of the number and **granularity** of the senses described.

A prominent example of a lexical database widely used in NLP is **WordNet** (Miller, 1995; Fellbaum, 1998). WordNet is a manually-built semantic network for English. Senses in WordNet are represented with **synsets**:[1] sets of (near-)**synonyms**, that is, words with the same (or highly

[1]WordNet synsets allow us to quickly verify the observation made in the previous Section (2.1) that words with multiple senses are used more frequently. 79% of lemmas in WordNet have a single synset, but their average frequency –as calculated on Google Ngrams (Brants and Franz, 2006)– is much lower (241.000) than that of words with more than one sense (7M).

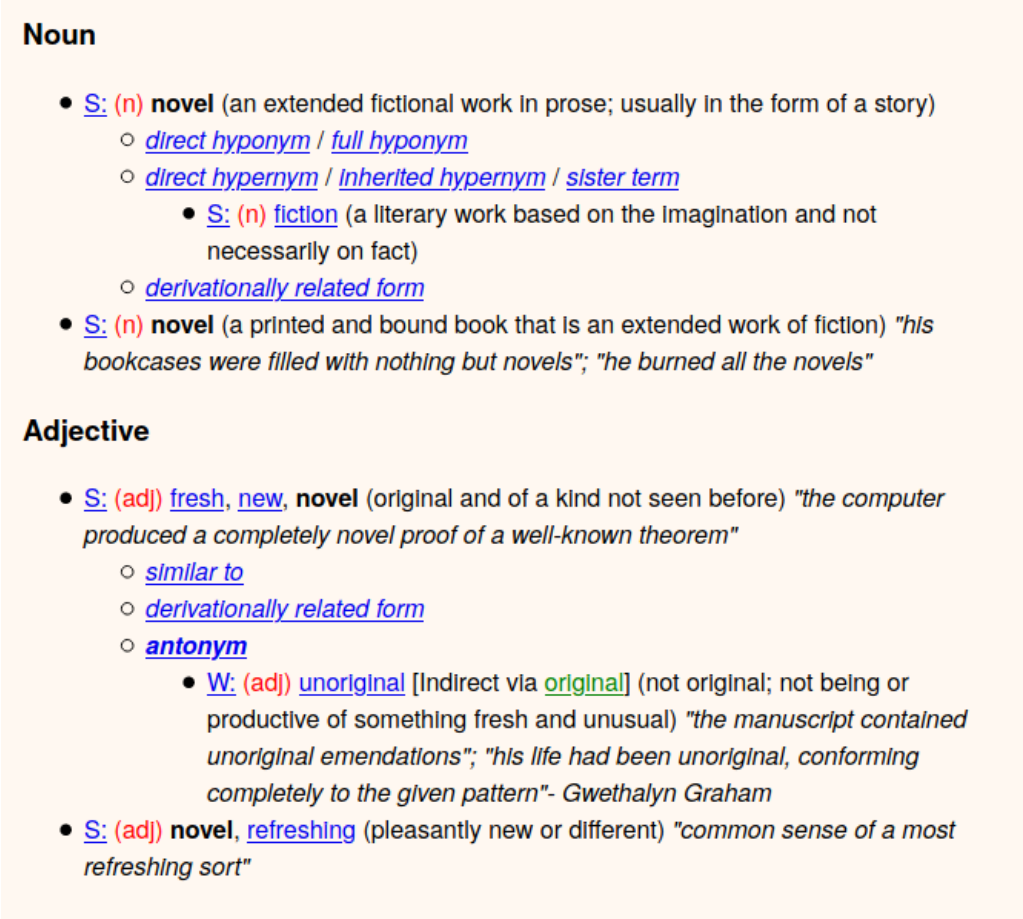
similar) meaning. Words with multiple senses (ambiguous or polysemous) are thus found in multiple synsets. Synsets are linked to other synsets in the WordNet hierarchy with which they stand in a particular semantic relation, such as hypernymy/hyponymy, troponymy, meronymy or antonymy. Additionally, synsets are often described with a short definition and sometimes contain usage examples for one or more of the words in it. Figure 2.2 shows the WordNet lexical entry for the word *novel*.

One of the commonly raised issues of WordNet is its granularity. As explained above, there is no unique solution to determining sense boundaries, and distinctions in WordNet tend to be very fine-grained for most NLP applications (Dolan, 1994; Palmer et al., 2004). For instance, we find 40 senses for the verb *run* in WordNet, whereas the online Cambridge Dictionary[2] lists 9 senses for it, with a few intra-sense distinctions. Other criticisms have made reference to its incomplete vocabulary, as it lacks specialised terms, named entities or neologisms (Smith and Fellbaum, 2004; McCrae et al., 2017). Despite this, WordNet is the de facto default sense inventory used in NLP for English. The biggest corpus with manual sense annotations, **SemCor** (Miller et al., 1993), with over 234,000 annotated word instances, uses the WordNet inventory.

The fact that it was manually created makes of WordNet a high-quality resource. It is however hard to create such high quality resources in other languages, or to extend existing ones in order to include new word senses and usages. WordNet-like resources have been proposed for other languages. Since building a separate resource (almost) from scratch is expensive in terms of time and effort, one common approach is to translate English WordNet into a target language. This was done in the EuroWordNet project (Vossen, 1998), which contains seven languages. The resulting resources can be directly compared to any other WordNet that preserves the English WordNet structure. At the same time, however, these resources are biased towards the English structure of the lexicon, which is rarely –if ever– fully compatible with that of other languages (Derwojedowa et al., 2008). Additionally, these databases tend to have a small coverage (Bond and Paik, 2012) and are in their majority automatically created, therefore they contain noise. **BabelNet** (Navigli and Ponzetto, 2012) is the biggest and highest-coverage WordNet-like resource. It is a semantic network where words in over 250 languages are organised into multilingual synsets. It was created automatically by joining the information present in WordNet and also Wikipedia, which served to include encyclopedic knowledge into the resource. It has later been extended with additional sources, such as WordNets in other languages.

In this thesis, we do not use senses to disambiguate word instances; we instead choose other ways of describing word meaning in a graded fashion (Section 2.1.3.1). We only use WordNet and BabelNet to retrieve the number of senses of words as an indication of their level of polysemy (Chapters 7 and 8). We also use SemCor in order to obtain data with controlled sense distributions (Chapter 7).

[2] https://dictionary.cambridge.org/dictionary/english/run



Noun

- S: (n) **novel** (an extended fictional work in prose; usually in the form of a story)
  - *direct hyponym* / *full hyponym*
  - *direct hypernym* / *inherited hypernym* / *sister term*
    - S: (n) fiction (a literary work based on the imagination and not necessarily on fact)
  - *derivationally related form*
- S: (n) **novel** (a printed and bound book that is an extended work of fiction) *"his bookcases were filled with nothing but novels"; "he burned all the novels"*

Adjective

- S: (adj) fresh, new, **novel** (original and of a kind not seen before) *"the computer produced a completely novel proof of a well-known theorem"*
  - *similar to*
  - *derivationally related form*
  - ***antonym***
    - W: (adj) unoriginal [Indirect via original] (not original; not being or productive of something fresh and unusual) *"the manuscript contained unoriginal emendations"; "his life had been unoriginal, conforming completely to the given pattern"- Gwethalyn Graham*
- S: (adj) **novel**, refreshing (pleasantly new or different) *"common sense of a most refreshing sort"*

Figure 2.2: Example of a WordNet entry (using WordNet's 3.1 online interface) for the word *novel*. It displays four synsets with their definition and, sometimes, example sentences. For one of its noun senses, we can see a hypernym synset (*fiction*). The picture also shows the antonym of one of its adjective senses (*unoriginal*).

2.1.3 Word Sense Disambiguation and Annotation

In this section, we describe three approaches for the semantic annotation of words: the use of word senses (Section 2.1.3.1), lexical substitutes (Section 2.1.3.2), and usage similarity (Section 2.1.3.3). We also describe resources and datasets used in this thesis for the last two approaches.

2.1.3.1 Word Sense Annotation

In the sense enumeration approach described in the previous section, the meaning of a word is often presented as a plain list of mutually exclusive word senses, which does not account for inter-sense relations. When a list of senses is used for word sense annotation, humans tend to show a low agreement (Krishnamurthy and Nicholls, 2000; Véronis, 1998; Murray and Green, 2004). This seems to improve with coarser-grained sense inventories (Palmer et al., 2007); and other factors like sense concreteness and specificity of the context also have an impact on annotator agreement (Passonneau et al., 2009); but some words are inherently difficult to disambiguate regardless of the inventory used, like *pull* (Palmer et al., 2007). Higher

		WordNet senses									
Word instance	**Annotator**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Snow covered areas appear *bright* blue in the image which was taken in early spring and shows deep snow cover.	Annotator 1	3	1	1	4	1	1	1	4	3	4
	Annotator 2	4	1	1	5	1	1	1	3	1	1

Table 2.1: Example of graded word sense annotation from the WSim dataset (Erk et al., 2009, 2013) for an instance of the word *bright*. The senses correspond to: 1-emitting light, 2-undimmed, 3-hopeful, 4-having a striking colour, 5-splendid, 6-happy, 7-intelligent, 8-having lots of light, 9-burnished, 10-reverberant. An annotation of 1 means the sense does not describe this instance of *bright* at all, and 5 that it perfectly corresponds to this instance.

polysemy (i.e. more senses), higher frequency, and a uniform sense distribution are also factors that contribute to a lower agreement (Martínez Alonso et al., 2015). Allowing the annotation of only one sense per word instance makes agreement even harder, especially in cases of underconstrained or sylleptic contexts, where multiple senses could apply (Jurgens, 2014).

These problems led to the development of graded annotation protocols, allowing annotators to propose multiple senses per usage (Véronis, 1998; Passonneau et al., 2012; Jurgens, 2013). Erk et al. (2009) propose a relaxation of the single-best-sense approach which consists in accepting multiple senses for a word instance, each to a different degree in a continuous scale. Given a word instance, they asked annotators to provide a graded judgment from 1 to 5 for each of its senses in WordNet indicating how well the sense describes its meaning. 1 means the sense does not apply, and 5 indicates that the sense describes the meaning perfectly. This annotation results in a distribution over possible senses for every word instance, instead of a single annotated sense, allowing for more subtle distinctions to be detected across usages of a word. Table 2.1 shows an example of this kind of annotation. The authors make three important remarks: first, that annotators made use of the full range of scores, which highlights the need for graded disambiguation, and the limitations of the single-best-sense approach. Second, they emphasise the higher agreement achieved on this task compared to previous annotation efforts. And third, that no consistent sense grouping could explain the obtained ratings, showing that this kind of graded annotation provides advantages that cannot be obtained with a coarser-grained sense inventory.

There are, however, other ways of describing the contextual variation of word meaning that can also reflect its continuous nature, without requiring a pre-defined inventory of discrete senses. An important advantage of not relying on a sense inventory is that it becomes easier to work with languages where such expensive resources might not be available.

2.1.3.2 Lexical substitutes as a proxy for meaning

One of the first alternatives proposed was to describe the meaning of specific word instances using in-context **lexical substitutes** (McCarthy, 2002), either in the same language (McCarthy and Navigli, 2007) or cross-lingually (Resnik and Yarowsky, 1999; Apidianaki, 2009; Mihalcea et al., 2010), with translations. When available, synonyms or near-synonyms of a word can be

used to describe the meaning of its instances in context. In the following example from the LexSub dataset (McCarthy and Navigli, 2007), the meaning of the two instances of the verb *think* is described by the available substitute annotations, which illustrate their difference in meaning (HAVING AN OPINION vs COGITATE):

(1) I *think* we should be allowed to pray for the grace to be victorious.
Substitutes: *believe, feel, be of the opinion, recommend*

(2) In the process of searching for the right combination to bring out that flavor, we *think*, we fail, we reflect, and hopefully, we succeed.
Substitutes: *consider, analyse, reason, contemplate*

Substitutes provide a graded representation of word meaning. The overlap of the sets of substitutes assigned to two instances reflects how similar the meaning of these instances is (Erk et al., 2009). See, for example, this other meaning of *think*:

(3) Shafer *thinks* we're going to cry.
Substitutes: *believe, feel, assume, reckon, suspect*

The meaning of the instance of *think* in (3) is similar to that expressed in example (1), which explains the partial overlap between their substitutes (*feel*, *believe*). Some substitutes differ because of the specific nuances expressed by the two instances: *be of the opinion/recommend* vs *assume/suspect/reckon*. The meaning expressed by *think* in (3) could be described as an opinion, like that in (1), but it also expresses a hypothesis, a guess. In contrast, (1) and (2) do not share any substitute because the senses expressed are clearly distinct. In what follows, we describe the lexical substitution datasets and related resources used in our experiments.

The first dataset with lexical substitute annotations was proposed in SemEval 2007, task 10 (McCarthy and Navigli, 2007). Data for this **LexSub** task were collected for 201 specific target words with balanced part of speech, and 10 sentences were selected from the Internet Corpus of English (Sharoff, 2006) for each of the words. Target words were chosen carefully so that words with different numbers of senses be represented. To alleviate the skewness often present in the frequency distribution of word senses,[3] the organisers manually selected the sentences for 79 of the words, forcing a more even sense distribution.

Concepts-in-Context (CoInCo) is another resource with substitute annotations. As opposed to LexSub, CoInCo contains substitute annotations for all words in a sentence. This results in a more natural frequency distribution of senses than in the LexSub dataset. CoInCo contains 2,474 sentences from the MASC corpus (Ide et al., 2008). It consists of 15,629 target instances for 3,874 unique target lemmas across different parts of speech. Instances were annotated with substitutes by crowd workers.

Table 2.2 contains examples from these two datasets. Other datasets with in-context substitutes exist (Sinha and Mihalcea, 2014; Biemann, 2013). We use LexSub because a subset of its sentences has additional semantic annotations (Section 2.1.3.3), and CoInCo for its bigger size

[3]The senses of a polysemous word tend to be unevenly distributed, with one or a few senses being much more commonly used than the rest (Kilgarriff, 2004). This is especially the case with frequently used words.

Sentence	Substitutes
LexSub	
We recommend that you **check** with us beforehand.	verify (3), confirm (2), report (1), make sure (1)
I have **checked** multiple times with my order and that is not the case.	verify (4), investigate (1), confirm (1), make sure (1)
The romance is uninspiring... and **dry**.	boring (2), uninteresting (2), dull (1), unsympathetic (1)
If the mixture is too **dry**, add some water; if it is too soft, add some flour	parched (2), unmoistened (1), desiccated (1), stodgy (1)
CoInCo	
A **mission** to **end** a **war**	**mission**: goal (2), plan (2), task (2), calling (1), campaign (1), dedication (1), devotion (1), duty (1), effort (1), initiative (1), intention (1), movement (1), pursuit (1), quest (1), step (1) **end**: stop (5), finish (4), conclude (2), halt (2), terminate (2), abolish (1), cease (1), **war**: fight (5), battle (3), conflict (3), combat (2), crusade (1), struggle (1)

Table 2.2: Example instances from two Lexical Substitution datasets: LexSub (McCarthy and Navigli, 2007) and CoInCo (Kremer et al., 2014)

and more natural distribution.

One resource particularly relevant for lexical substitution is the **Paraphrase Database (PPDB)** (Ganitkevitch et al., 2013; Pavlick et al., 2015),[4] a large collection of paraphrase pairs available in multiple languages (Ganitkevitch and Callison-Burch, 2014). It was automatically built using the pivot method (Bannard and Callison-Burch, 2005), which discovers paraphrases by finding expressions that share a translation in bilingual parallel corpora. For instance, the fact that *aim* and *goal* share the French translations *objectif* and *but* is taken as an indication that *aim* and *goal* share some meaning and are, therefore, paraphrases of each other. PPDB contains paraphrases at the word as well as the phrase level. The paraphrases in English PPDB were later automatically ranked by quality based on human judgments (Pavlick et al., 2015), creating PPDB 2.0. The English PPDB was also automatically enriched with entailment relations (e.g. EQUIVALENCE – the relation holding between *airport* and *aerodrome*; FORWARD ENTAILMENT – as in *airport* ⇒ *facility*) and stylistic information in the form of formality and complexity scores (for example, the difference in formality between *father* and *daddy* is bigger than that between *kids* and *children*). The English PPDB contains over 80 million paraphrase pairs and 140 paraphrase patterns. Their ranking by quality has served as a criterion to split the database into multiple paraphrase packages of different sizes (from S to XXXL), ranging from highest precision (smallest size) to highest recall (biggest size). Figure 2.3 contains the first 30 paraphrases for the word *novel* in PPDB 2.0 XXL. Most of them reflect its noun and adjective senses, but there are also a few incorrect entries.

In our work, we use substitutes as a way to approximate word meaning. We evaluate the ability of contextualised representations to propose lexical substitutes in context using the

[4] http://paraphrase.org

continued, inserted, innovative, added, novelty, fiction, novelist, newer, new,developments, romance, new, changes, book, nouvelle, original, unpublished, groundbreaking, tale, roman, narration, story, non-conventional, nouveaux, non-traditional, fresh, pioneering, brand-new, emerging, unconventional

Figure 2.3: 30 first paraphrases for the word *novel* in the Paraphrase Database 2.0 XXL. A stronger colour indicates the paraphrases that are contained in smaller packages (L and XL).

LexSub dataset (Chapter 3) and we then use automatic substitute annotations to complement, and in some cases enrich, the representations (Chapters 4, 5 and 6). We use CoInCo as additional training data for usage similarity estimation (Chapters 4 and 6). Finally, we also use paraphrases at the word level from the PPDB as candidate substitutes when performing lexical substitution (Chapters 4, 5 and 6).

2.1.3.3 Usage similarity

Erk et al. (2009) also consider the notion of **usage similarity**, or similarity between two instances of the same word, to account for the graded distinctions between word instances. For example, the similarity between the instances of *think* in the sentences (1) and (3) above would be higher than the similarity between the instances in sentences (1) and (2).

In this section we introduce several datasets that address in-context word similarity, both between different words and between usages of the same word. Table 2.3 contains examples extracted from these datasets.

Usim (Erk et al., 2009, 2013)[5] is a dataset which contains 10 instances for each of 56 lemmas manually annotated with graded pairwise usage similarity judgments. Each sentence pair received a rating (on a scale of 1-5, from less to more similar) by multiple annotators, and the average judgment for each pair was retained. Word instances are taken from the LexSub dataset (McCarthy and Navigli (2007), Section 2.1.3.2), adding an extra layer of semantic annotation. This kind of data allows to study the organisation of the semantic space of individual words without comparing them to other words.

Word-in-Context (WiC) (Pilehvar and Camacho-Collados, 2019) consists of 7,466 pairs of contextualised instances for the same target word. In this case, the task is framed as a binary classification, where instances describe either the same or a different sense, instead of being in a similarity continuum. WiC sentences were extracted from example usages in WordNet (Fellbaum, 1998), VerbNet (Schuler, 2006) and Wiktionary[6] and were automatically labelled using information available in these resources. Meanings represented in the WiC dataset are generally coarser-grained than WordNet senses, which was ensured by excluding WordNet synsets describing highly similar meanings. The human-level performance upper-bound on this binary task is 80.5%. It was calculated as the average accuracy of four annotators on 100-instance samples of WiC. Inter-annotator agreement is also high, at 79%. This dataset has been

[5] http://www.dianamccarthy.co.uk/downloads/WordMeaningAnno2012/

[6] https://www.wiktionary.org/

Label/ Score	Sentence 1	Sentence 2
	Stanford Contextual Word Similarity (SCWS)	
8/10	world 's heroes fight them off . Most of the X-Men die , but Iceman (alongside Rogue , Storm , Colossus , and Jean Grey) is able to survive Magneto 's attack . He is last seen demolishing the X-Mansion alongside Rogue and Jean Grey and burying the **deceased** X-Men in its place . He finds it hard to destroy their home , but he feels it to be the right thing to do now that Professor Xavier is dead . In the first story arc of Ultimate Comics Spider-Man , the Post-Ultimatum version of Ultimate Spider-Man ,	which may act as a safeguard against rising waters or predators , or as a method of regulating humidity and temperature) . The male takes no part in caring for its young , and retreats to its year-long burrow . The female softens the ground in the burrow with **dead** , folded , wet leaves and she fills the nest at the end of the tunnel with fallen leaves and reeds for bedding material . This material is dragged to the nest by tucking it underneath her curled tail . The female Platypus has a pair of ovaries but
	WiC	
T	Laws limit the **sale** of handguns .	They tried to boost **sales**.
F	She didn't want to **answer**.	This may **answer** her needs.
	CoSimLex	
6.96 (9.50-2.54)	The intercoastal trip took about 17 days each **way** and the ships called at either Los Angeles or San Diego on eastbound and westbound trips. With two ships on the route, one ship departed from either New York or San Francisco about every three weeks. The service was marketed as the ideal **manner** to visit the Panama-California Exposition in San Diego and the Panama-Pacific International Exposition in San Francisco. *Similarity: 2.54*	However, the true burden of the tax cannot be properly assessed without knowing the use of the tax revenues. If the tax proceeds are employed in a **manner** that benefits owners more than producers and consumers then the burden of the tax will fall on producers and consumers. If the proceeds of the tax are used in a **way** that benefits producers and consumers then owners suffer the tax burden.. *Similarity: 9.50*
	Usim	
4.3/5	We recommend that you **check** with us beforehand.	I have **checked** multiple times with my order and that is not the case.
1.3/5	The romance is uninspiring... and **dry**.	If the mixture is too **dry**, add some water; if it is too soft, add some flour.

Table 2.3: Example instances from each dataset addressing word similarity in context.

used on a shared task (Espinosa-Anke et al., 2019), in which we participated (Chapter 4), which addresses the similarity estimates that can be derived from contextualised representations. It has also been included in the SuperGLUE Benchmark (Wang et al., 2019a), a battery of challenging tasks that aim to measure a model's overall level of language understanding.[7] The current best model,[8] T5 (Raffel et al., 2020), obtains a 76.9 score, approaching the human upper-bound. There is also a multilingual version of the WiC dataset, XL-WiC, which was created in a similar way to WiC using Multilingual Wordnet and Wiktionary, and is available in 12 languages (Raganato et al., 2020).

A similar dataset, but more focused on the similarity between instances of different words, is the **Stanford Contextual Word Similarity (SCWS) dataset** (Huang et al., 2012). It was initially designed to evaluate sense embeddings (Section 2.2.3.2). It contains 2,003 sentence pairs manually annotated by 10 crowdworkers with similarity scores from 0 to 10. Most

[7] https://super.gluebenchmark.com/

[8] As of September 14th, 2020

sentence pairs in SCWS compare different words, but some instances compare different senses of the same target word. Sentences were extracted from Wikipedia and automatically selected to trigger specific senses of a word. Pilehvar and Camacho-Collados (2019) note, however, that the inter-rater agreement on this dataset is very low (average pairwise Spearman's $\rho = 0.35$).

Another recently created dataset that addresses word similarity in context is **CoSimLex** (Armendariz et al., 2020a). CoSimLex differs from WiC in several aspects: it contains graded, not binary, judgments; it compares instances of different words, not of the same word; and it is available in several languages: English, Croatian, Finnish and Slovene. In contrast with Usim, WiC and SCWS, an instance consists of a single short text snippet containing the two target words to compare. Annotators had to provide similarity judgments for the two words in their shared context. Every target word pair is present in two contexts, allowing to assess the effect of context on the perceived similarity between the two words. Word pairs were extracted from Simlex-999 (Hill et al., 2015) and its translations, and the sentences come from each language's Wikipedia. Contexts with different degrees of similarity were pre-selected using two contextual models, ELMo (Peters et al., 2018a) and BERT (Devlin et al., 2019) (Section 2.2.3.3). An expert annotator made the final context selection to be included in the dataset.

Along with lexical substitutes, usage similarity has a central role in this tesis as a way of accounting for word meaning. Specifically, in order to evaluate the lexical semantic quality of contextualised representations, we investigate how well they reflect words' usage similarity (Chapter 4). With the same goal, we also use the similarity between usages of different words (Chapter 6). Additionally, we explore whether usage similarity estimations from the representations reflect whether they are ambiguous, polysemous, or vague (Chapter 5, (McCarthy et al., 2016)); and their level of polysemy (i.e, their number of senses (Chapter 7)). In our experiments, we use the Usim, WiC and CoSimLex datasets, for their higher quality.

2.2 Vector Space Models of Word Meaning

We have seen that many words are polysemous, and their meaning varies across contexts. We have presented different ways of accounting for this variation: using lists of senses, lexical substitutes and usage similarity annotations. In this section we focus on computational approaches to word meaning which create vector representations for words. We first present the underlying principle of all these approaches, the Distributional Hypothesis (Section 2.2.1). Then, we introduce traditional Vector Space Models which build representations relying on co-occurrence counts from corpora (Section 2.2.2). Finally, we describe several models that learn representations with language model (LM) objectives (Section 2.2.3). We describe models that represent words at different levels (at the type-, sense- or token- level), and also present ways in which these representations can be evaluated.

2.2.1 The Distributional Hypothesis

Lexical semantics is the area of linguistics that studies the meaning of words. There is no single way to define the notion of **word meaning**: multiple theories from disciplines such as

philosophy, linguistics or cognitive science characterise it in different ways; for example as an abstract mental representation (Rosch, 1975; Lakoff and Johnson, 2008), or through the use of minimal conceptual building blocks or "semantic primitives" (Katz and Fodor, 1963; Wierzbicka, 1972). In this thesis, we adopt a **distributional** point of view. The Distributional Hypothesis (Harris, 1954), often illustrated with Firth's (1957) famous quote "You shall know a word by the company it keeps", states that "difference of meaning correlates with difference of distribution". In other terms, two words with different meanings appear in different contexts, while two semantically **similar** words tend to occur in the same contexts.

It is important to distinguish the notion of semantic similarity between words from **synonymy** and **relatedness**. Two words are synonyms if they are equivalent in meaning, i.e. if they mutually entail each other (Kreidler, 1998). Absolute synonyms are interchangeable: one word can be replaced by the other without affecting the truth conditions of a sentence (Cruse, 1986).[9] Two words are said to be semantically related if they are associated in some way, for example by means of meronymy, a part-whole relation (as *leg* and *person*) or by a function relation (e.g. *teeth* and *toothpaste*, or *vet* and *dog*) (Budanitsky and Hirst, 2006).

An example of similar words would be *cat* and *dog*, whose meanings share common traits. According to the Distributional Hypothesis, their similarity is reflected in the fact that they are very often used in the same contexts:

(4) I just fed the [cat|dog].

(5) I took my [cat|dog] to the vet.

However, we know *cat* and *dog* do not have exactly the same meaning (i.e. are not synonyms) because there are also contexts that they do not share:

(6) His [dog] barks when it's hungry.

From a distributional point of view, the meaning of a word is determined by its **similarity** to other words. Similarity between words is, in turn, defined by the number of contexts shared between them.

Cat and *dog* are said to be in a paradigmatic relation, because they can often occupy the same position in sentences (i.e. they often co-occur with the same words), whereas *vet* and *dog* stand in a syntagmatic relation, as they often co-occur with each other (i.e. they are used in the same sentence) (Schütze and Pedersen, 1993). This distinction is useful in order to tell apart the notions of similarity and relatedness. Related words are not necessarily semantically similar, and often stand in a syntagmatic relation (Turney, 2008).

The link between the Distributional Hypothesis of meaning and textual data has allowed for its empirical corroboration in studies of human perception of semantic similarity (Rubenstein and Goodenough, 1965; Miller and Charles, 1991), where it was found that distributional similarity correlates with human judgments. With the increasing computational power and text digitalisation of the last two decades, this idea has greatly influenced the field of computational

[9] As opposed to absolute synonyms, "partial" synonyms or near-synonyms are highly similar in meaning, but differ in some aspects, typically connotational, such as style (*dad* and *father*), emotion or intensity (*good* and *great*).

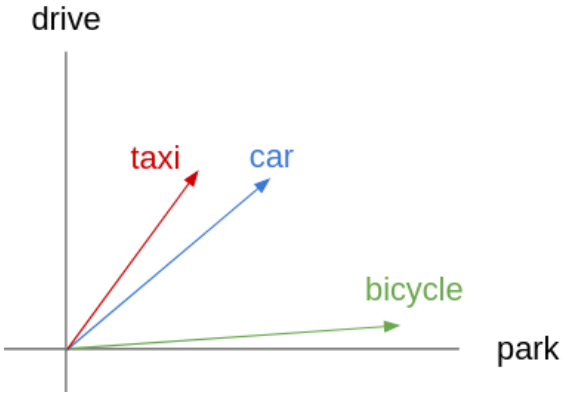


Figure 2.4: Two-dimensional example of distributional vectors. *Taxi* and *car* often co-occur with *drive* and with *park*, whereas *bicycle* rarely co-occurs with *drive* (one *rides*, but doesn't *drive*, a bicycle). The cosine similarities between the vectors (which rely on the angle between them) reflect that *taxi* and *car* are more similar to each other than either of them is to *bicycle*.

semantics and is the underlying principle behind many word representation approaches. In the following sections, we describe models that build vector representations based on the distributional hypothesis.

2.2.2 Distributional Approaches to Word Meaning

Distributional approaches seek to obtain word representations (in the form of vectors) reflecting the semantic similarity between words. These vectors are created from co-occurrence data, and encode different kinds of information obtained from the different contexts in which a word occurs in a corpus. The definition of **context** is highly parametrizable: it can be a fixed-size window surrounding the word, the sentence containing it, or even a document where the word appears. It can simply take into consideration other words that occur in the context (the bag-of-words approach) or it can use additional linguistic information from syntactic annotations (Padó and Lapata, 2007; Baroni and Lenci, 2010; Levy and Goldberg, 2014a). Note that with the term "context" (of a word), we refer to linguistic information surrounding, but not including, the target word instance that is to be represented.

The obtained vectors configure a semantic space. Distributionally similar words (which share co-occurrence patterns) have vectors that are close in the space. The semantic similarity of words (which correlates with distributional similarity) can thus be calculated with different measures of vector distance (or similarity), such as the Euclidean distance or the widely used cosine similarity. Using these measures, one way of characterising the meaning of a word is observing the words that are closest to it in the space, in other words, retrieving its nearest neighbours. Figure 2.4 shows a simplified example of word vectors in the space created by these models.

The first distributional word vectors were based on Vector Space Models (VSMs) (Salton et al., 1975; Turney and Pantel, 2010; Baroni and Lenci, 2010). The features (dimensions) of these vectors correspond to meaningful units, such as other words or documents. Their values

indicate, for example, the frequency of co-occurrence of words in a corpus, or the presence of words in documents. These "explicit" (Levy and Goldberg, 2014b) representations are high-dimensional and sparse, but can be compressed with dimensionality reduction techniques (Landauer and Dumais, 1997), at the cost of interpretability.

What most VSM approaches have in common (Lund and Burgess, 1996; Bullinaria and Levy, 2007; Padó and Lapata, 2007) is the representation of word **types**, i.e. every word is represented with a single vector, regardless of whether it is polysemous or monosemous. There have been, however, different proposals to account for polysemy. One of the solutions proposed is to create multiple vectors per word, corresponding to their different senses (Schütze, 1998; Pantel and Lin, 2002; Reisinger and Mooney, 2010; Van de Cruys and Apidianaki, 2011). The first work in this direction was by Schütze (1998), who proposed a method for Word Sense Induction (Manandhar et al., 2010; Jurgens and Klapaftis, 2013), i.e. for discovering word senses from text in an unsupervised way. The approach consists in representing the context of a word instance using the centroid of the vectors of the words in it. These context vectors are then clustered based on their proximity in the semantic space, and the resulting clusters are assumed to represent different word senses.

Another solution to account for polysemy is using vectors that represent words in context. The first approaches of this type were focused on **semantic composition**, and typically consisted in combining type-level vectors of words in a phrase. Semantic composition is very relevant for capturing word meaning in context, since the meaning of a word instance strongly relies on its neighbouring words. For example, the phrases *football match* and *perfect match* evoke distinct meanings of *match*. The goal of these approaches was to represent a complex expression (e.g., a multi-word phrase or a sentence). The simplest model of composition represented a sentence as the average of the vectors of the words in it (Landauer and Dumais, 1997). Kintsch (2001) and Mitchell and Lapata (2008) studied other composition operations to combine the meanings of two words, such as addition and multiplication. Erk and Padó (2008) and Thater et al. (2009, 2010, 2011) built upon this work, enriching phrase representations with syntactic information such as selectional preferences and dependency relations. These works are focused on building representations of phrases taking word-type vectors as the point of departure. Erk and Padó (2010) and Reddy et al. (2011a) instead represented word **tokens** by proposing an **exemplar-based** model that does not rely on sense or word type vectors in any stage of the process. In this case, the goal was to represent word instances in context (or the context surrounding these instances).[10] A word type is represented as a set of instance vectors, some of which are activated to form an instance representation adapted to the new context of use. In contrast to models of composition, these exemplar-based models are not limited to combining two words, and take information from the whole sentence into account to represent a word instance. These first studies typically evaluated representations on tasks such as lexical substitution (McCarthy and Navigli, 2007), in- and out-of-context word similarity estimation,

[10]While models of composition aim at representing a complex expression (for example, the phrase *football match*), exemplar-based approaches obtain a representation of an instance of *match* in its sentential context (Baroni et al., 2014a).

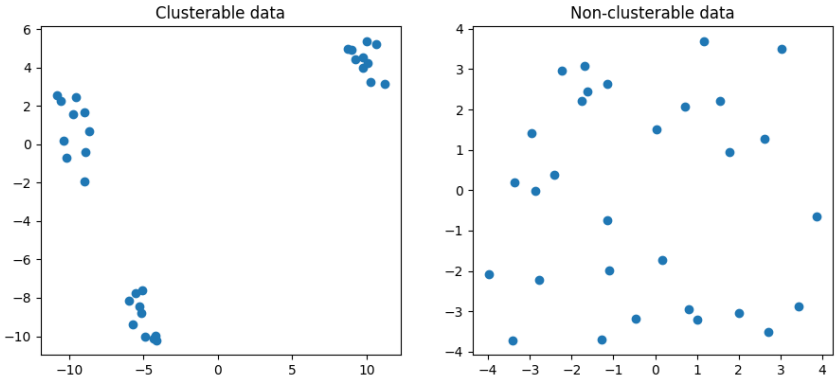


Figure 2.5: Artificial examples of a highly clusterable dataset (left) and a dataset with no cluster structure, i.e. non-clusterable (right).

word sense disambiguation (Thater et al., 2011) and compositionality detection (Reddy et al., 2011b), where they showed improvements over previous word type representations.

Representations at the token level offer a way not only to represent the contextual variation of word meaning, but also to explore the ambiguity-vagueness spectrum of words in the semantic space. A first effort in this direction was that of McCarthy et al. (2016). They propose the notion of the **partitionability** of a word into senses; that is, the ease with which the senses of a word can be distinguished. A word with clearly distinct senses (e.g. *coach*, or *bank* with its FINANCIAL INSTITUTION and RIVER BANK senses) is easier to partition into senses than a word with vague semantics (e.g., *thing*, whose meaning can subtly vary in every context of use). They use word usage similarity annotations from the Usim dataset (Erk et al. (2009, 2013), Section 2.1.3.3) to determine the actual partitionability of a word: if the instances of a word received many mid-range similarity scores (between 2 and 4 in a scale from 1 to 5), or if a word presented a low inter-annotator agreement, they assume that its semantic space is harder to partition into senses. McCarthy et al. (2016) propose a computational method to create vectorial representations of word usages from substitute and translation annotations, and estimate partitionability in terms of the clusterability of the obtained representations. Clusterability is a notion from the machine learning literature that measures the extent to which a set of data points have an inherent cluster structure (Ackerman and Ben-David, 2009; Adolfsson et al., 2019). If a dataset is not clusterable or has low clusterability, one should not proceed with clustering, as results could be misleading. Figure 2.5 shows examples of a clusterable and a non-clusterable dataset. In this thesis, we build upon their work and try to predict the partitionability of words using token-level word representations from modern contextual language models (Section 2.2.3.3) as a way of evaluating their lexical semantic knowledge (Chapter 5).

2.2.3 Distributed Approaches to Word Meaning (Word Embeddings)

Vectorial word representations have evolved and improved in many respects in the last few years, becoming an essential part of virtually any NLP system. Work by Bengio et al. (2003), Collobert and Weston (2008) and later Mikolov et al. (2013b,a) constituted a big leap forward, introducing **predictive** models (Baroni et al., 2014b). Instead of gathering co-occurrence counts from corpora (as in the **count-based** approaches introduced in the previous section), these models essentially merge the tasks of language modelling and representation, or **embedding**, learning. The language modelling (LM) task typically consists in predicting a word given a context. The definition of context is, again, not fixed: in traditional language models (statistical LMs and unidirectional recurrent neural networks), the context consists of the words occurring only before the target word to be predicted. In other models, the context is made of the words in a window surrounding the target word, or the whole sentential context of the word. The distributional knowledge required to solve this task is learned and at the same time encoded in the dense representations built for the words, and these seem to be better than count-based models at reflecting word meaning and human judgments of semantic similarity (Baroni et al., 2014b), despite depending more heavily on the right hyper-parameter choice (Levy and Goldberg, 2014b). These language models rely on different types of neural network architectures and the specific training objective used varies for each model. In this section, we focus on the progression from predictive approaches that assign a single vector to a word type (Mikolov et al., 2013a; Pennington et al., 2014), or **static**, type-level approaches, to models that propose multiple representations for a word. Of the latter, one can distinguish between those that propose a representation for every sense of a word (**multi-prototype** or **sense** representations (Neelakantan et al., 2014; Iacobacci et al., 2015)), and the recently developed **contextual** models of word representation (Peters et al., 2018a; Devlin et al., 2019), which are able to assign a different vector to every new usage of a word. These token-level, contextualised representations are the focus of our work. For a thorough survey of word embedding methods we refer the reader to Camacho-Collados and Pilehvar (2018).

2.2.3.1 Static Embeddings

Mikolov et al. (2013a)'s word2vec is probably the most well-known word embedding approach. It is a neural model which efficiently learns dense representations of words from large amounts of data with a language model objective. Vectors can be built using two architectures: continuous bag of words (CBOW) and Skip-gram. In CBOW, the model is shown an averaged representation of context words and has to predict a target word that appeared in this context. In the Skip-gram architecture, the task is the inverse: the model receives a target word as input and must predict the words that appear in its context. Table 2.4 shows an example illustrating the difference in the two tasks.

These two approaches are proposed alongside two strategies that contribute to the model's training speed and to the quality of the resulting representations: "subsampling of frequent words" and "negative sampling". Subsampling frequent words consists in assigning words a

Sentence	**Approach**	**Training samples** (input / desired output)
She *reads* a book	CBOW	(*she, a, book / reads*)
	Skip-gram	(*reads / she*), (*reads / a*), (*reads / book*)

Table 2.4: Example of training instances used by the CBOW and Skip-gram word2vec models (Mikolov et al., 2013b).

probability of being deleted from the training corpus depending on their frequency. Specifically, the higher the frequency of a word, the higher the probability of deleting it. Highly frequent words (especially stop words like *the* or *a*) often contribute very little to the meaning of other words in the sentence, and they would constitute a big portion of the training examples without subsampling. Negative sampling is crucial to the model's speed. For each training instance, a limited number of negative (incorrect) words are selected, and only the weights for these words are updated, instead of weights for all words in the vocabulary.

Many subsequent approaches build on, or are inspired by, word2vec, such as FastText (Bojanowski et al., 2017), which incorporates character information for more morphology-aware representations that can better encode rare words. The model is based on Skip-gram, but words are represented as a sum of character n-gram embeddings. GloVe embeddings (Pennington et al., 2014) combine the advantages of count-based and predictive models, arguing that the latter do not make use of global co-occurrence statistics from a corpus. The model mixes local context information (as used in word2vec) with global co-occurrence data. Levy and Goldberg (2014a) adapt Skip-gram to make use of syntactic contexts with dependency parsing. Doc2vec (Le and Mikolov, 2014) is also based on word2vec, but extends it to create representations of sentences and documents.

Despite their success and good performance on many NLP-related tasks (Zou et al., 2013; Baroni et al., 2014b; Passos et al., 2014), these static word embeddings are, by definition, incapable of accounting for the different meanings of ambiguous or polysemous words. Just as in type-level representations from VSMs, polysemous and monosemous words are equally represented with a single embedding, meaning that all senses of a polysemous word are conflated into a single representation. This has inevitable consequences on the resulting semantic space, where the vectors of semantically dissimilar words like *pollen* and *refinery* are found close to each other because they are both related to (different senses of) the word *plant* (Neelakantan et al., 2014).

2.2.3.2 Multi-prototype and Sense Embeddings

The meaning conflation problem of type-level embeddings motivated research on representations of lexical meaning that could account for polysemy. Among the proposed solutions are multi-prototype and sense embeddings, which correspond to different word senses. In this case, a given word type has a finite number of representations available, one of which can be chosen to represent a word instance in context. **Sense embeddings** are linked to an external sense inventory, such as WordNet or Wikipedia (Iacobacci et al., 2015; Camacho-Collados

et al., 2015; Rothe and Schütze, 2015; Pilehvar and Collier, 2016). These embeddings can be learnt by using sense definitions (Chen et al., 2014) or sense-annotated corpora (Iacobacci et al., 2015). **Multi-prototype embeddings** induce senses from corpora evidence alone, directly or through static word embeddings (Pelevina et al., 2016). This can be carried out in a "two-stage" process (Camacho-Collados and Pilehvar, 2018): an initial sense induction step followed by the creation of embeddings for each of the induced senses (Huang et al., 2012; Liu et al., 2015). In other neural-based methods, sense induction and embedding learning are performed simultaneously (Neelakantan et al., 2014; Tian et al., 2014; Li and Jurafsky, 2015). Sense and multi-prototype embeddings generally improve results on out-of-context word similarity tasks, and they are more suitable to estimating in-context word similarity or usage similarity (Huang et al., 2012) than static representations.With sense and multi-prototype embeddings, this can be done by first assigning an embedding to the specific instances (Li and Jurafsky, 2015), or by weighting the similarity according to the probability of each sense (Reisinger and Mooney, 2010; Huang et al., 2012; Chen et al., 2014). Li and Jurafsky (2015) evaluate sense embeddings on several NLP tasks, and identify a few tasks where they provide an advantage over static embeddings (word and sentence similarity, semantic relation identification and part of speech tagging), and others where they do not help (e.g. sentiment analysis). At the same time, however, they find that simply increasing the dimensionality of static embeddings can provide similar gains on these tasks.

This type of embeddings constitute an important advancement towards a more realistic way of representing word meaning which accounts for polysemy. However, as discussed in Sections 2.1.2 and 2.1.3.1, a list of discrete senses –and the corresponding sense embeddings– falls short to represent the meaning nuances between instances of words with vague semantics. Additionally, given the difficulty to determine the number of senses for a word, the initial models which were not based on external lexical resources made the strongly simplifying assumption that all words have the same number of senses (Huang et al., 2012; Tian et al., 2014). Some alternatives were proposed later (Neelakantan et al., 2014; Bartunov et al., 2016) which induce the number of senses from corpus data. Finally, an important downside of sense embeddings is that their integration into NLP models is not as straightforward as that of static word embeddings, since it requires an additional, preliminary disambiguation step.

2.2.3.3 Contextualised Embeddings

The next breakthrough came with **contextualised word embeddings**, such as ELMo (Peters et al., 2018a) and BERT (Devlin et al., 2019). Instead of representing word types or word senses, contextualised embeddings represent word tokens (instances). They differ from multi-prototype embeddings and sense embeddings in that they assign to every word instance an embedding that is specific to its context of use, and which does not come from a finite list of (sense-)embeddings. They thus have potential to describe the subtle meaning nuances expressed by different word instances. These models have given unprecedented performance in multiple NLP tasks such as Question Answering and Natural Language Inference.

In Section 2.2.2, we have introduced the first distributional token-level representations and

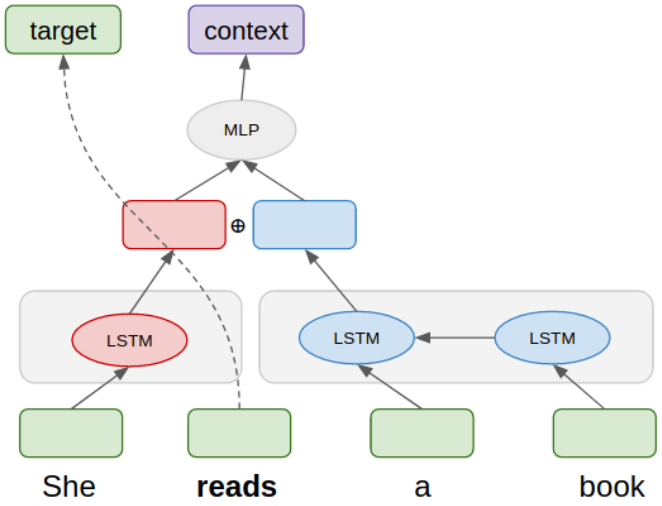


Figure 2.6: Diagram of the context2vec architecture. The context of a word (*reads*) is encoded with a left-to-right LSTM and a right-to-left LSTM, followed by a non-linear layer (Multilayer Perceptron, MLP). The Figure is inspired by Melamud et al. (2016)'s Figure 1 (b).

models of composition. Recent contextualised approaches rely instead on neural language models. One of the first and very influential neural contextualised models is **context2vec** (Melamud et al., 2016). This model does not produce word instance representations, but it generates embeddings for sentential contexts in the same space as static word embeddings, and is optimised to reflect inter-dependencies between them. context2vec uses a neural network architecture based on word2vec's CBOW (Mikolov et al., 2013a). It replaces CBOW's representation of a word's surrounding context (consisting of a simple average of the embeddings of the context words in a fixed window) with a neural representation of the context obtained using a bidirectional Long Short-Term Memory (biLSTM). Figure 2.6 illustrates the architecture of this model.

Peters et al. (2018a)'s **ELMo** (Embeddings from Language Models) relies on a bidirectional LSTM (biLSTM) (Hochreiter and Schmidhuber, 1997; Graves and Schmidhuber, 2005). that is trained with a language model objective on a large corpus to obtain deep contextualised word representations. ELMo representations are deep in the sense that they are a linear combination of all the internal layers of the model. ELMo can be integrated into task-specific architectures, where the task and the linear combination of different layers are simultaneously learned in a supervised way. Alternatively, representations can be extracted from the model and used separately. The ELMo model is illustrated in Figure 2.7 (left). The original model consists of three layers: a first, character n-gram convolutional layer followed by two biLSTM layers.

BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) is also a language model, which uses a Transformer architecture (Vaswani et al., 2017). Instead of using a forward and a backward language model separately like in ELMo, BERT jointly conditions on the left and right context in all layers. It is trained with a double pre-training objective: Masked Language Modelling (MLM) and Next Sentence Prediction (NSP). MLM is

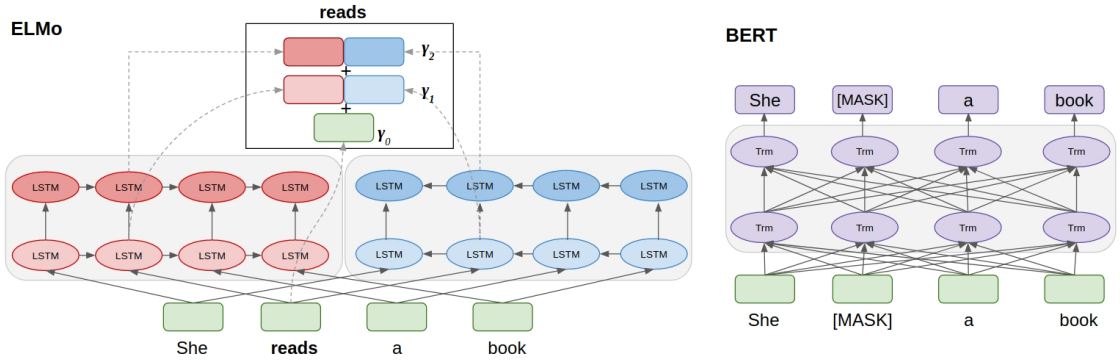


Figure 2.7: Simplified diagrams representing the architectures of the ELMo and BERT models. ELMo has a non-contextualised character-based input layer followed by two layers of left-to-right and right-to-left LSTMs. The final embedding of a word in context is a linear combination of the representations in the three layers. BERT uses a deep (12- or 24-layer) Transformer (Trm) architecture and directly outputs contextualised word embeddings, although it is also possible to use word embeddings from the hidden layers. The Figure is inspired on Devlin et al. (2019)'s Figure 3.

equivalent to a *Cloze* task (Taylor, 1953) that consists in predicting, given the whole (left and right) context, a random word that has been masked. During pre-training, 15% of words have to be predicted: 80% of them are replaced with a special [MASK] token, 10% with a random token, and another 10% are left unchanged. BERT is trained with a specific kind of tokenisation based on sub-word units, called "wordpieces" (Schuster and Nakajima, 2012; Wu et al., 2016). A wordpiece vocabulary V with a pre-specified size |V| is generated from a training corpus minimising the number of word splits done. This generally results in dedicated vocabulary items (tokens) for the most common words in the training corpus, while less frequent words are split into multiple wordpieces, which do not necessarily correspond to morphemes.

Another distinctive feature of BERT is the use of the special tokens [CLS] and [SEP]. [CLS] marks the beginning of the input sequence and serves as a classification token aggregating information from the whole sequence. Classifiers for tasks at the sequence level take this token as input. [SEP] marks the end of a segment, and separates two sentences for the NSP pre-training task. For example, an input sequence could be "[CLS] She [MASK] a book . [SEP] It is a novel . [SEP]". The input embedding to BERT for a given wordpiece is the sum of its corresponding token embedding, an embedding marking the position it occupies in the input sequence, and a segment embedding indicating whether it belongs to the first or the second sequence (up to or after the first [SEP] token). In the example above, the input embedding for *She* would be a sum of the embedding for the *she* wordpiece, an embedding for tokens at the 2nd position, and an embedding common to all words in the first segment (*[CLS], she, [MASK], a, book, [SEP]*).

BERT provides a unified architecture that can be fine-tuned on data for different tasks with the simple addition of a classification or regression head, without the need of having a task-specific architecture as in ELMo. The BERT architecture is illustrated in Figure 2.7 (right).

BERT has inspired a whole generation of Transformer-based language models. Lighter

versions of BERT (with fewer parameters) have been proposed, such as DistilBert (Sanh et al., 2019) or Albert (Lan et al., 2020). Other models change the training objectives: RoBERTa (Liu et al., 2019b) is an optimised version of BERT that is not trained with the NSP task, and XLNet (Yang et al., 2019) replaces the MLM objective with a permutation-based LM objective. In T5 (Raffel et al., 2020), all tasks are turned into a text-to-text format, and the tokens at the beginning of the sequence indicate the task that the model has to perform (i.e. "summarize : ..."). There also exist multilingual models trained on text in different languages, such as XLM (Conneau and Lample, 2019) and the multilingual version of BERT (Devlin et al., 2019). Many monolingual versions of BERT for languages other than English have also been proposed (e.g. Flaubert (Le et al., 2020) and CamemBERT (Martin et al., 2020) for French, BETO for Spanish (Cañete et al., 2020) or GreekBERT (Koutsikakis et al., 2020)). Our experiments with context-sensitive representations involve mainly context2vec, ELMo and different BERT models. We compare their performance to that of static embeddings.

These models have motivated a large body of research work on identifying the knowledge that is encoded in them and the representations they generate. This **interpretability** work aims at finding information in the models and investigating the reasons behind their high performance across benchmarks. We expand on the topic of interpretability and probing of contextualised models in Section 2.3.

2.2.3.4 Word Embedding Evaluation

The quality of a type of word meaning representation can be determined by evaluating them extrinsically or intrinsically. An extrinsic evaluation consists in incorporating them into a pipeline to solve an NLP task and assessing their contribution to the results obtained. Embeddings have been used for various tasks such as Part of Speech tagging, Named Entity Recognition (Ghannay et al., 2016; Ma and Hovy, 2016; Lample et al., 2016), Sentiment Analysis (Kim, 2014) and Neural Machine Translation (Qi et al., 2018; Artetxe et al., 2018). In this thesis we focus on intrinsic evaluation, which is often done with word similarity tasks. In this case, what is evaluated is whether the similarity between word embeddings in the semantic space correlates with human judgments of word (type) similarity. Ideally, if representations encode distributional knowledge, they should reflect the semantic similarity between words. We distinguish two kinds of intrinsic evaluation: out-of-context and in-context.

Out-of-context intrinsic evaluation Numerous datasets exist for out-of-context (word type) similarity, which contain human judgments of semantic similarity (and/or relatedness) for word pairs: RG-65 (Rubenstein and Goodenough, 1965), MC-30 (Miller and Charles, 1991), WS-353 (Finkelstein et al., 2001) and its split into similarity and relatedness pairs (Agirre et al., 2009), Mturk-287 (Radinsky et al., 2011), Mturk-771 (Halawi et al., 2012), RW (Luong et al., 2013), SimLex-999 (Hill et al., 2015), and more. This kind of evaluation has been criticised, among others, for the subjectivity of the judgments and the low correlation with extrinsic evaluation results (Faruqui et al., 2016; Chiu et al., 2016). Crucially, these datasets are unable to account for polysemy due to their lack of context. They can serve for the evaluation of

static embeddings, but they are not enough for testing the more fine-grained lexical semantic knowledge of sense embeddings and contextualised embeddings.[11]

Other tasks proposed for the out-of-context intrinsic evaluation of word embeddings also focus on word similarity, but re-frame the question. Some examples are synonymy detection (Baroni et al., 2014b), where the model has to choose the best synonym for a target word among a limited number of options; word analogy solving (Mikolov et al., 2013b), consisting in finding the 4th term of a semantic- or syntactic-based analogy (*dog* is to *puppy* what *cat* is to _____) or outlier word detection, that is, identifying a word that deviates from the rest of words in a specific set (Camacho-Collados and Navigli, 2016).

In-context intrinsic evaluation In-context tasks are more adequate for the evaluation of multi-prototype, sense and contextualised embeddings, as the model needs to make use of the context to generate the representation for a word instance. Static embeddings can also be evaluated on in-context tasks, but in a less straightforward way –for example, a representation for a word instance can be obtained by averaging its embedding and those of the words surrounding it. The simplest way of performing this kind of evaluation is through an in-context similarity task, which can involve usages of the same word or instances of different words. The datasets introduced in Section 2.1.3.3 can serve to this end: Usim (Erk et al., 2009, 2013) for graded usage similarity, WiC (Pilehvar and Camacho-Collados, 2019) for binary usage similarity, SCWS (Huang et al., 2012) for graded usage and word instance similarity, and CoSimLex (Armendariz et al., 2020a) for graded word instance similarity in multiple languages. Datasets annotated with lexical substitutes (Section 2.1.3.2) can also be used for evaluation. The lexical substitution task, which consists in selecting meaning-preserving substitutes for words in context, was initially proposed as a testbed for Word Sense Disambiguation systems (McCarthy and Navigli, 2007), but in recent work it is mainly seen as a way of evaluating the in-context lexical inference ability of vector-space models without explicitly accounting for sense (Kremer et al., 2014; Melamud et al., 2015, 2016). Models can be evaluated for their ability to propose and/or rank substitutes for a word in context. In this thesis, we precisely evaluate several context-sensitive representations on lexical substitution and in-context word similarity tasks, using the LexSub (McCarthy and Navigli, 2007), CoInCo (Kremer et al., 2014), Usim, WiC and CoSimLex datasets (Chapters 3, 4, 6).

Finally, another kind of intrinsic evaluation, which we do not explore in this thesis, is Word Sense Disambiguation (WSD). The quality of the representation of meaning in word embeddings can be evaluated by including them in a WSD system and assessing their ability to correctly assign senses to words in context. Different kinds of embeddings have been applied to this task: static (Iacobacci et al., 2016; Taghipour and Ng, 2015), sense embeddings (Chen et al., 2014; Rothe and Schütze, 2015) and contextualised representations (Reif et al., 2019;

[11] In fact, with sense embeddings, the similarity of two polysemous words out of context is typically defined as the similarity between their two most similar senses (Camacho-Collados et al., 2015; Mancini et al., 2017). Thus, *coach* and *bus* would be considered to be very similar, but a static embedding model may assign them a lower similarity because it conflates the two main senses of *coach* (BUS and TRAINER) in the same representation (Faruqui et al., 2016).

Wiedemann et al., 2019; Loureiro et al., 2020).

2.3 Interpretability Studies

We present an overview of the recent body of work on the interpretability of contextual word embedding models. These studies aim at unraveling the knowledge contextual models acquire during pre-training. In Section 2.3.1 we introduce the methodology used in these studies, and in Section 2.3.2 we describe the main findings about the (lexical) semantic knowledge encoded in these models.

2.3.1 Interpretability Methods

The recent developments in deep language models like BERT (Devlin et al., 2019), ELMo (Peters et al., 2018a) or GPT-2 (Radford et al., 2019) brought about an interest in understanding the kind of linguistic and world knowledge that they learn during pre-training. Importantly, in multi-layer models of this type, the question that is investigated is not only what information the models encode, but also where this information is located. This field has come to be known as **interpretability** or **BERTology**, because most of these studies are focused on the BERT model. While contextual pre-trained LMs obtain outstanding results in numerous NLP tasks, their complexity makes it hard to understand what their good performance is due to. Analysing these "black boxes" promotes a better understanding of their inner workings and of the information that they encode, and can provide important insights as to how they can be improved. Examining the predictions of these models can also be useful for identifying weaknesses in evaluation datasets, and promoting the design of challenging tasks that cannot be solved with simple heuristics (McCoy et al., 2019).

In what follows, we present an overview of this line of work. This is a recent and rapidly evolving field. We refer the reader to Rogers et al.'s (2020) "primer on BERTology" for a thorough review of the main outcomes from studies focused on the BERT model.

Interest in the interpretability of NLP models started with the transition from traditional VSMs to deep, neural network models (Belinkov and Glass, 2019). Early studies would focus on discovering information encoded in static word embeddings such as word2vec (Mikolov et al., 2013a) and GloVe (Pennington et al., 2014) (Köhn, 2015; Gupta et al., 2015; Ettinger et al., 2016); in neural sentence representations (Adi et al., 2017; Conneau et al., 2018) or in the hidden representations of recurrent neural networks (Karpathy et al., 2015; Shi et al., 2016; Li et al., 2016; Hupkes et al., 2018). We focus on the latest developments in this field, aimed at analysing contextualword embedding models like BERT and ELMo and other models relying on the Transformer architecture (Vaswani et al., 2017).

One of the most common approaches for analysing deep language models is the use of **probing classifiers**. A probe (also called a "diagnostic classifier" (Hupkes et al., 2018)) is typically a classifier that uses representations from the model that is being studied as input. The probe is trained on a task of interest, for example part of speech (PoS) tagging, and its performance on this task is taken as an estimation of how much information the model encodes

about the kind of knowledge necessary to solve the task. For example, Tenney et al. (2019b) develop a set of probes on multiple linguistic tasks (PoS tagging, dependency labeling, semantic role labeling (SRL) and coreference, among others) using the same kind of classifier for all tasks: a 2-layer binary multi-layer perceptron (MLP) that takes as input span-based representations of CoVe (McCann et al., 2017), OpenAI (Radford et al., 2018), ELMo and BERT. They find that all models tend to perform better than non-contextualised baselines, especially on syntactic tasks. Hewitt and Manning (2019) learn a linear transformation of ELMo and BERT representations at different layers where the L^2 distance between two words reflects their distance in a parse tree, and find an impressive amount of syntactic knowledge in the representations which varies across layers. Liu et al. (2019a) also probe the representations of three models (ELMo, BERT and OpenAI) at different layers. They test them on 17 different linguistic tasks, such as Named Entity Recognition (NER), Grammatical Error Detection, or syntactic dependency arc prediction. Their focus is on the transferability of representations; i.e. how general (as opposed to task-specific) representations at different layers of a model are, as reflected in their performance on linguistic tasks different than the pre-training task. They conclude that the first layer of LSTMs is the most transferable, as layers are increasingly task-specific. Transformers do not exhibit the same trend: their most transferable representations are located in the middle layers.

Importantly, Liu et al. (2019a) raise the question of how complex a probe should be. The more complex it is, the we rely on representations from the original model that is being evaluated. Precisely, they obtain better performance in some tasks when simply increasing the classifier complexity (from a linear model to a MLP). This is one of the criticisms that have been raised about probing: the fact that a probe cannot uncover a certain type of linguistic knowledge does not mean the knowledge is not present (Tenney et al., 2019b). At the same time, Tenney et al. (2019b) and Hewitt and Liang (2019) note that the fact that a probing classifier obtains good performance does not reveal how, or if, the models use the linguistic knowledge that they are probed for. To solve this problem, Hewitt and Liang (2019) advocate for the use of "control tasks", where each word type is assigned a random label. This type of tasks can help identify reliable probing classifiers: a probe that learns the underlying word-label mapping in a control task is not insightful, as its performance on a real task could simply be due to its ability to memorise such patterns. On the other hand, a probe that makes use of the linguistic information encoded in representations is expected to perform better in a real task than in a comparable control task. Alternatively, Voita and Titov (2020) propose to quantify, from an information-theoretic perspective, the "amount of effort" needed to learn a certain task. The intuition behind their approach is that if representations encode a specific kind of information, they can be trained to transmit it using fewer bits.

Another popular interpretability approach, concretely for models trained with an MLM objective, relies on fill-in-the-gap or **cloze-style** tasks (Petroni et al., 2019; Ettinger, 2020). These tasks evaluate the language model capabilities of an MLM and require no additional training. They consist in querying the MLM for a missing token in a set of cloze statements designed to target a specific kind of information. For example, to probe BERT for world or

encyclopedic knowledge, and concretely for birthplaces, we can create the statement "Shakespeare was born in [MASK]". The MLM produces a ranking of the words that could fill this slot ordered by probability. The position of the correct word in the ranking by probability is used for evaluation. The higher the correct word is in the ranking, the better the information is considered to be encoded in the model. Goldberg (2019) uses this kind of probing to investigate BERT's syntactic knowledge. The author investigates subject-verb agreement by checking, for instance, the relative probabilities of *is* and *are* in sentences like "the game that the guard hates [MASK] bad". Talmor et al. (2020) test BERT and RoBERTa (Liu et al., 2019b) in a number of multiple-choice tasks that involve symbolic reasoning, such as the comparison of two numeric values, or that of the size of different objects. To account for answers that are made of multiple wordpieces, they propose a supervised approach using QA statements (Talmor et al., 2019), where the first segment of the input is a question ("What is usually located at hand and used for writing?") which is concatenated with each of the possible answers as a second segment (after the first [SEP] token), one at a time. They find that, overall, the models are strongly context-dependent and incapable of abstract reasoning. Petroni et al. (2019) propose to use fill-in-the-blank statements to probe BERT for factual and common-sense knowledge. They compile the LAMA (LAnguage Model Analysis) benchmark, a set of cloze-style prompts built from knowledge triples (BAILEY PENINSULA, *located in*, `Antarctica`) or question answer pairs.[12] LAMA contains an extensive number of relations including birthplaces, locations, consequences ("Sometimes virus causes [MASK]"), company products ("iPOD Touch is produced by [MASK]"), or prerequisites ("Typing requires [MASK]"). They find that for some types of relation, an off-the-shelf BERT model pre-trained with the MLM objective is comparable to other dedicated methods relying on oracle knowledge.

One downside of the fill-in-the-gaps approach is the model's sensitivity to slight changes in the prompts used. Ravichander et al. (2020) probe BERT for the hypernymy relation and find BERT's predictions to be inconsistent across prompts using singular and plural (e.g. "a car is a [MASK]" vs "cars are [MASK]"). Similarly, Jiang et al. (2020) propose modifications to cloze statements in LAMA and demonstrate their impact on the results.

There has also been extensive work on analysing **self-attention weights** in the Transformer network (Raganato and Tiedemann, 2018; Voita et al., 2019b). These studies analyse the attention heads in all layers of the model, looking for patterns in the tokens they attend to. Clark et al. (2019) examine BERT and localise a number of attention heads that seem to be specialised in certain linguistic notions related to syntax and coreference, such as the object of verbs or co-referent mentions. They also find that numerous attention heads exhibit the same behaviour, with many of them focusing on the [SEP] token. Similarly, Kovaleva et al. (2019) identify a small number of attention patterns that are repeated in multiple attention heads. Using different methodology, Kovaleva et al. (2019), Voita et al. (2019b) and Michel et al. (2019) show that it is posible to prune or disable several attention heads at test time in different

[12]The dataset contains statements illustrating relations between entities stored in Wikidata, common sense relations between concepts from ConceptNet (Speer and Havasi, 2012), and knowledge aimed at answering natural language questions in SQuAD (Rajpurkar et al., 2016).

Transformer-based models without causing a big loss in performance, which is a symptom of the over-parametrisation of these models.

2.3.2 Semantic Knowledge in Pre-trained Language Models

Most of the early interpretability studies on contextualised representations addressed grammatical and syntactic aspects of language, such as part of speech (Hewitt and Manning, 2019; Hewitt and Liang, 2019), subject-verb agreement (Goldberg, 2019) and function words (Kim et al., 2019). The first studies addressing semantics explore phenomena in the syntax-semantics interface, such as semantic role labelling and coreference (Kovaleva et al., 2019; Tenney et al., 2019a; Liu et al., 2019a; Clark et al., 2019; Peters et al., 2018b). Tenney et al. (2019a) observe that, for BERT, the best layers for these two tasks are located in the upper half of the Transformer model, while syntactic tasks are better solved in earlier layers. For other semantic tasks (semantic relations and proto-roles) information seems to be spread quite evenly across layers.

Lexical meaning has recently started attracting increasing attention, and has been the object of several interpretability studies. Some of these focus on word meaning at the type level, while other works explore how these models handle word sense distinctions. We also include in our overview studies which, without directly addressing lexical meaning, provide some interesting insights about the type-level information inside the model.

The fact that contextualised embeddings represent word instances offers a convenient way to explore how they deal with aspects of word meaning that are related to contextual variation, but there is no straightforward way to investigate the type-level knowledge they contain. To explore the knowledge these models have about lexical meaning at the word type level, Vulić et al. (2020) and Bommasani et al. (2020) propose different ways of obtaining a static (type-level) embedding from contextualised word representations, for example by aggregating the representations of a word across multiple contexts or by feeding a word in isolation into BERT. This complements a strand of work that investigates how contextualised and static representations can benefit from each other. In these works, contextualised embeddings are used to train a static embedding model (Wang et al., 2019b) or are combined with static embeddings (Akbik et al., 2019; Liu et al., 2020). Bommasani et al. (2020) evaluate the lexical semantic knowledge in different Transformer-based models (BERT, RoBERTa, GPT-2, XLNet (Yang et al., 2019) and Distilbert (Sanh et al., 2019)) using tasks such as out-of-context word similarity and word analogies, and report consistent improvements over purely static representations like word2vec (Mikolov et al., 2013a) and GloVe (Pennington et al., 2014). Vulić et al. (2020) compare monolingual and multilingual BERT models on these and other out-of-context tasks, such as Bilingual Lexicon Induction (BLI). One of their findings is that monolingual models encode lexical information of higher quality than multilingual models. Importantly, they carry out an analysis by layer and conclude that lexical knowledge is spread throughout multiple layers of BERT models, but is particularly present in the lower layers. This contrasts with Tenney et al. (2019a)'s observation, highlighting the differences between the semantic tasks addressed.

Another study that sheds light on the location of lexical information in contextual LMs, albeit not directly addressing word meaning, is that of Ethayarajh (2019). This work explores

precisely the degree of contextualisation in token-level representations extracted from BERT, ELMo and GPT-2 at different layers. The author investigates the similarity estimates that can be drawn from these representations, which serve as an indication of how context-specific they are and provide useful observations regarding the impact of context on the representations. One of the most remarkable findings are the highly distorted similarities obtained. They are due to the anisotropy of the vector representations, which occupy only a narrow cone in the space. This issue affects all tested models but seems to be extreme in the last layers of GPT-2, resulting in highly similar representations even for random words. The author also observes that representations are more contextualised in the top layers, although contextualisation is not monotonic. This is consistent with Voita et al. (2019a)'s findings, who investigate how different pre-training tasks (Masked Language Modelling (MLM), traditional LM, and Machine Translation (MT)) influence the flow of information in the Transformer architecture. Adopting an information-theoretic point of view, they estimate the mutual information between a token representation at a certain layer and the input token. An interesting observation is that while with the LM and MT pre-training objectives information about the input token is monotonically lost across layers, with an MLM objective this information is initially lost, but is recovered at the last layer just before prediction. They call these different phases in MLMs "context encoding" and "token reconstruction". The fact that the higher layers are the most contextualised, and contain less information about the input token, could explain why lower layers are better at type-level lexical tasks.

Some works use cloze-style queries to probe BERT for type-level lexical semantic knowledge. Ravichander et al. (2020) report that BERT encodes knowledge about hypernymy better than static methods, but its performance strongly depends on the prompt used. In a supervised setup, Bouraoui et al. (2020) fine-tune BERT for several relations, including lexical ones (e.g., meronymy, synonymy, antonymy, collective nouns and light verb constructions) and find that BERT performs better on tasks requiring encyclopedic knowledge than on lexical semantics tasks, where they obtain mixed results.

Other work studying the lexical knowledge in BERT looks at how word instance representations reflect the different senses of words. Wiedemann et al. (2019) and Reif et al. (2019) propose experiments using representations built from Wikipedia and the SemCor corpus (Miller et al., 1993), and observe that BERT can organise word usages in the semantic space in such a way that reflects the meaning distinctions present in the data. They further demonstrate BERT's disambiguation capacity by means of supervised experiments on the word sense disambiguation (WSD) task. Reif et al. (2019) additionally explore how word meaning in BERT representations is affected by context. They observe that when a sentence c_1 containing a word w used in a specific sense s is concatenated (through the conjunction *and*) with another sentence c_2 containing w used in a different sense s', the embedding of w in c_1 moves towards the centroid of s'. This results in a decrease in WSD performance, and highlights BERT's high sensitivity to context.

The interplay between lexical and contextual information in the hidden representations of LSTM LMs has also been explored. Aina et al. (2019) propose to train diagnostic classifiers on

the tasks of retrieving the input embedding of a word and a representation of its contextual meaning (as reflected in its lexical substitutes). Their results show that both types of knowledge (lexical and contextual) seem to be present to varying degrees at different layers and hidden states. Other works address usage similarity in contextual LMs by evaluating them on WiC (Pilehvar and Camacho-Collados, 2019) and on CoSimLex (Armendariz et al., 2020b), or investigate how "distributional" a model is, i.e. whether the similarities derived from its representations reflect the expected semantic distributional similarities (Mickus et al., 2020).

The work carried out in this thesis contributes to our understanding of the information encoded in contextual language models. Our experiments, and particularly our analyses by layer, provide valuable insights as to how well the models encode knowledge about different lexical semantic aspects, and where this knowledge is located. We believe that a better understanding of what models like BERT are capable of, and of their limitations, can help to trace directions for improvement.

Chapter 3

In-context Lexical Substitution

3.1 Introduction

As explained in Chapter 2, in-context lexical substitutes are a way of describing a word's meaning without recurring to word senses. The lexical substitution task consists in selecting candidates to substitute a word instance, and ranking them according to their appropriateness in a given context. For example, *virus*, *insect* and *error* are all possible substitutes of the word *bug*. However, when used in a specific context (e.g. "I'm sick with the stomach *bug*"), only some substitutes are acceptable (i.e. *virus*). A model that is able to use the semantic information provided by the context should thus rank *virus* over *insect* and *error* in this specific sentence.

The importance of context in defining the meaning of word instances and selecting the substitutes that best fit specific sentences makes of this task an ideal testbed for contextualised representations. These representations model complex characteristics of word usage, and give state-of-the-art performance in a variety of NLP tasks involving syntactic and semantic processing (Peters et al., 2018a; Devlin et al., 2019).

In this chapter, we present our work investigating the lexical substitution capability of different context-sensitive word and context representations, including context2vec, ELMo and BERT. Each model accounts for context in a different way. We want to learn how well different types of representations, with various underlying architectures and training objectives, are able to encode word meaning in context. The quality of the substitute ranking proposed by a specific type of representations is taken as an indication of the model's ability to capture the semantic information necessary for the lexical substitution task.

We compare these representations on the SemEval 2007 Lexical Substitution task dataset (McCarthy and Navigli, 2007), LexSub, using existing similarity-based unsupervised methods. Additionally, we experiment with a way to tune these context-sensitive representations to sense-specific contexts of use (Cocos and Callison-Burch, 2019) and explore the impact of this tuning on the lexical substitution task. We also compare the performance of contextual models to baseline models that exploit static word embedding representations for measuring semantic similarity without directly accounting for context, such as GloVe (Pennington et al., 2014) and FastText (Mikolov et al., 2018).

Sentences	Substitutes
The panther **fired** at the bridge and hit a truck.	shoot (5)
While both he and the White House deny he was **fired**, Frum is so insistent on the fact that he quit on his own that it really makes you wonder.	sack (5), dismiss (1)
As a **coach**, we speak and listen with the intent of helping people surface, question and reframe assumptions.	trainer (3), teacher (2), instructor (1), tutor (1)
We hopped back onto the **coach** - now for the boulangerie!	bus (5), carriage (1)

Table 3.1: Examples of manually proposed substitutes for the verb *fire* and the noun *coach* in the SemEval-2007 Lexical Substitution dataset (McCarthy and Navigli, 2007). Numbers in brackets indicate the number of annotators who proposed each substitute.

Our results shed light on the semantic quality of different contextualised representations, and highlight the importance of the architecture and objectives used for model training in capturing information relevant for the lexical substitution task.

3.2 The Task

The Lexical Substitution task can be decomposed into two steps: (1) collecting candidate substitutes, and (2) ranking the candidates according to how well they fit in a given context. In our experiments, as in previous work (Erk and Padó, 2008; Thater et al., 2010; Apidianaki, 2016), we focus solely on the ranking task: systems are not expected to identify substitutes from the whole vocabulary, but rather to estimate the suitability of items in a specific pool of substitutes and rank them accordingly. The set of candidate substitutes $S_t = \{s_1, s_2, ..., s_n\}$ for a target word t used in our experiments consists of all the paraphrases proposed for t across all its instances in the LexSub dataset. In Table 3.1, we present examples of substitutes for words in context proposed by annotators in the SemEval-2007 Lexical Substitution dataset. Substitutes are ranked by the number of annotators who proposed them.

Early approaches to solve this task used type-level representations and consisted in adapting the representation of a word to each specific context of use. This was done by combining the basic vector of the word with the vectors of words found in its immediate context, or standing in some syntactic relation with it (Erk and Padó, 2008; Thater et al., 2010, 2011). Substitutes were considered to be appropriate if their representations were similar to this contextualised representation.

3.3 Data

In this section we describe the dataset used to evaluate the models, LexSub (McCarthy and Navigli, 2007) and a resource used to tune the representations to specific target-substitute (t, s_i) pairs, PSTS (Cocos and Callison-Burch, 2019).

Substitutes	PSTS sentences
sack	Yet what are proclamations on employment rights worth, when company bosses have a 'divine right' to hire and **fire**?
dismiss	They chose to **fire** a lot of people; to throw people out who weren't needed.
shoot	We hope that the generals and civilian oligarchs will not **fire** on the honduran people.
launch	A security source said electrical wiring found at the site suggested plans to **fire** the rockets by remote control.

Table 3.2: Examples of PSTS sentences for the verb *fire* corresponding to each one of its candidate substitutes (*sack, dismiss, shoot* and *launch*).

The LexSub Dataset As explained in Section 2.1.3.2, the LexSub dataset was proposed in SemEval 2007, task 10 (McCarthy and Navigli, 2007). It contains 2100 sentences, ten for each of 210 target words. Five annotators were asked to provide at most three substitutes per word instance, avoiding multi-word expressions when possible. The number of annotators that proposed a specific substitute determines the gold substitute ranking. We evaluate different lexical substitution methods on the LexSub test set. This subset contains 1710 sentences for 171 target words. To ensure all methods are evaluated in the same conditions, we use a filtered version of the test set including 168 target words and 1,584 sentences. More details about the filtering procedure are given in Section 3.5.

Paraphrase-Sense-Tagged Sentences (PSTS) As we have seen with the *bug* example in the Introduction (Section 3.1), different synonyms or paraphrases might reflect different senses of a target word. The PSTS dataset (Cocos and Callison-Burch, 2019) provides sentences corresponding to different paraphrases of words, and thus to their different senses. Specifically, PSTS contains 10,000 example sentences for each of 3 million target-substitute pairs (t, s_i), where the target word is used in the sense described by the substitute s_i. This dataset was automatically compiled based on paraphrase pairs from the Paraphrase Database (PPDB, (Ganitkevitch et al., 2013; Pavlick et al., 2015), Section 2.1.3.2). Sentences come from the same English-to-foreign bitext corpora used to generate English PPDB. Examples for a (t, s_i) pair are sentences where the aligned translation t' of the target t (e.g. the French term *ver* for the English word *bug*) is also a possible translation of s_i (e.g. *worm*). Sentences are ranked by quality based on how characteristic the translation t' is of t. In Table 3.2, we give examples of PSTS sentences for the target word *fire* used in the senses described by its candidate substitutes (*sack*, *dismiss*, *shoot*, *launch*). This resource can be useful for lexical substitution, as it groups sentences where a target word appears with the meaning of one of its paraphrases. One of our experiments aims to see whether incorporating this fine-grained substitute information into our models can improve performance. For this purpose, we build representations based on the sentences provided in the resource (Section 3.4.1).

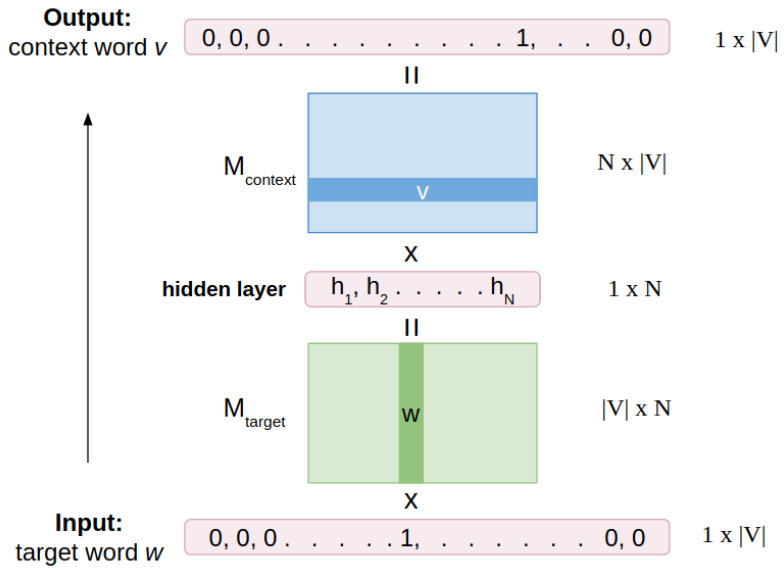


Figure 3.1: Skip-gram architecture. M_{target} is the embedding matrix typically used to represent words. $M_{context}$ contains embeddings of words as context elements. $|V|$ is the vocabulary size and N is the size of the hidden layer.

3.4 Experimental Setup

3.4.1 Context-sensitive Representations

In our experiments, we use context-sensitive word and context representations generated by different models. Each model accounts for context in a different way depending on the underlying architecture and training objective.

- **Skip-gram** (Mikolov et al., 2013a; Melamud et al., 2015). word2vec's Skip-gram model learns two distinct representations for every word type, one as a target and another as a context, both embedded in the same space. This is illustrated in Figure 3.1. The word-as-context representations are considered internal to the model and are generally discarded after training, and the output word embeddings represent context-insensitive target word types. Melamud et al. (2015) proposed to explicitly leverage the word-as-context embeddings generated within skip-gram in conjunction with the word-as-target embeddings to model word instances in context. The vectors used by Melamud et al. (2015) are syntax-based embeddings created with *word2vecf* (Levy and Goldberg, 2014a). We use the lighter adaptation proposed by Apidianaki et al. (2018) which circumvents the need for syntactic analysis, and use 300-dimensional skip-gram word-as-target and word-as-context embeddings trained on the 4B words of the Annotated Gigaword corpus (Napoles et al., 2012).

- **ELMo** (Peters et al., 2018a). In ELMo, word vectors are learned functions of the internal states of a deep bidirectional language model (biLM). The model contains three layers, so each token in text has three different representations, one per layer. It is important

to note that we do not train or fine-tune the ELMo model for lexical substitution, so we do not learn a linear combination of the biLM layers in the way ELMo is typically used. Instead, we experiment with the top layer (*ELMo-top*) and an average of the three layers (*ELMo-avg*) of the biLM (5.5B) trained on 5.5B tokens from Wikipedia and news crawl data, released by Peters et al. (2018a).[1] The representations from this model are 1024-dimensional.

- **context2vec** (Melamud et al., 2016) learns a generic context embedding function using a biLSTM network. We use 600-dimensional embeddings from a context2vec model trained on the ukWaC corpus (Baroni et al., 2009).[2]

- **BERT** (Devlin et al., 2019) is a deep Transformer language model trained with a cloze task objective. we use the `bert-base-uncased` (Devlin et al., 2019) model, trained on 3.3B tokens from BooksCorpus (Zhu et al., 2015) and Wikipedia, and extract representations from the top layer (*BERT-top*) and the average of the last four layers (*BERT-avg (4)*).[3] When a word is split into multiple word pieces, we average the representations of all its pieces.

We also use the sets of sentences available for each target-substitute pair in PSTS to create ELMo and context2vec representations for candidate substitutes. For ELMo, we use the approach proposed by Peters et al. (2018a) for applying the biLM representations to a supervised word sense disambiguation (WSD) task. More precisely, a representation for a substitute $s_i \in S_t$ of a target word t is the average of the ELMo vectors obtained from PSTS sentences corresponding to this (t, s_i) pair. For each substitute, we use the top-ranked 100 sentences, avoiding sentences with a high overlap in words.[4] We again use the top layer (*PSTS-ELMo-top*) and an average of the three layers (*PSTS-ELMo-avg*) of the 5.5B ELMo model.

For context2vec (*PSTS-c2v*), we create context representations from the sentences retained for a target-substitute (t, s_i) pair by replacing the target word with a blank slot. A representation for the substitute s_i is then created by taking the average of all generated context representations.

Figure 3.2 illustrates how we obtain substitute embeddings from PSTS sentences with ELMo and context2vec. The obtained candidate vectors are used in the lexical substitution methods described in Section 3.4.2.

3.4.2 Lexical Substitution Methods

Given an instance of a target word t in a context C and a set of candidate substitutes $S_t = \{s_1, s_2, ..., s_n\}$, each model provides a ranking of the substitutes depending on how well they

[1] https://allennlp.org/elmo

[2] https://github.com/orenmel/context2vec

[3] When the experiments described in this chapter were carried out, the BERT model had only recently appeared. We later applied it to the Lexical Substitution task using one of the best-performing methods (*tTs*, Section 3.4.2) and report the obtained results here.

[4] We use an overlap threshold of 60%. This cleaning serves to discard highly similar sentences and ensure a varied vocabulary in the retained dataset. If for some substitutes less than 100 sentences are available after this filtering, we retain them all.

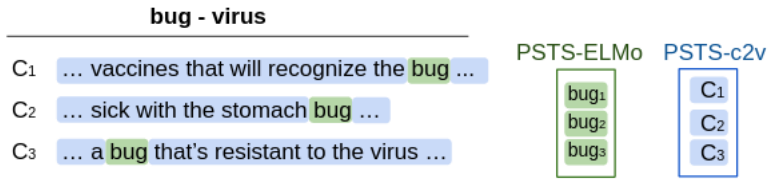


Figure 3.2: Depiction of the embeddings derived from PSTS for the target word *bug* used in its *virus* sense. We use word instance embeddings for ELMo (*PSTS-ELMo-top/avg*) and context vectors for context2vec (*PSTS-c2v*).

describe the meaning of t in C. Higher ranked substitutes should be (a) good paraphrases of the target, and (b) a good fit in the context.

In what follows, we describe how the different methods represent words and contexts, and how they use these representations to perform substitute ranking for every word instance. An illustration of the different methods can be found in Figure 3.3.

3.4.2.1 Target-to-substitute similarity (*tTs*)

ELMo and BERT representations are contextualised, meaning that the embedding that is generated for a token is a function of the full sentence in which it appears, and therefore it already contains information from the surrounding context. We propose a substitute ranking method that uses solely target-to-substitute (*tTs*) similarity, as measured by the cosine similarity of the corresponding ELMo/BERT representations.

Given a new context C of an instance of the target word t to be substituted, we first obtain an ELMo/BERT representation corresponding to t in C. Then, we replace t with all its potential substitutes in S_t, one at a time, as shown in Table 3.3, and obtain the vector for each substitute s_i in the context C by feeding the new sentence as input to the model. Substitutes are then ranked by the cosine similarity of the target word's vector in C with that of the vector of each substitute s_i in the same context.

They chose to **fire** a lot of people; to throw people out who weren't needed.
They chose to **sack** a lot of people; to throw people out who weren't needed.
They chose to **dismiss** a lot of people; to throw people out who weren't needed.
They chose to **shoot** a lot of people; to throw people out who weren't needed.
They chose to **launch** a lot of people; to throw people out who weren't needed.

Table 3.3: Substitution procedure to obtain contextualised candidate substitute representations from ELMo and BERT. In the *tTs* method, the vector of *fire* in this sentence is compared to those of *sack, dismiss, shoot* and *launch* in the same context.

We use this method with PSTS-ELMo as well. For each context C, possible substitutes in S_t are ranked according to the similarity of their PSTS-ELMo embedding (obtained from PSTS) to the ELMo embedding of the target word t in C.

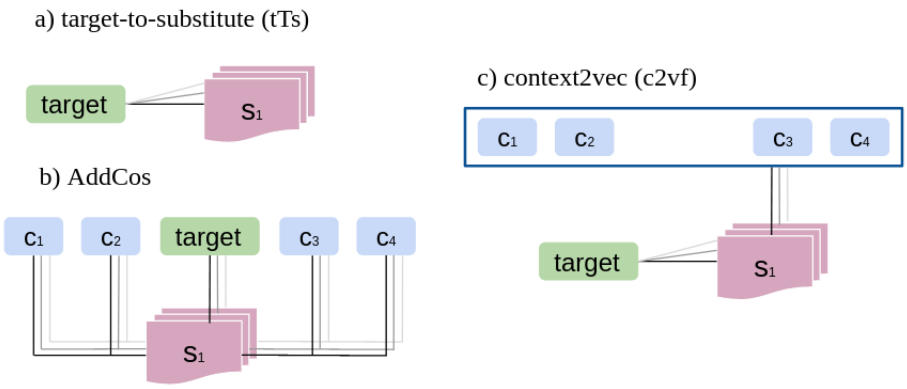


Figure 3.3: Illustration of the type of context information the different methods use: a) *tTs* uses target to substitute similarity only (Section 3.4.2.1); b) *AddCos* also uses similarities between a candidate and each of the words in the surrounding context (Section 3.4.2.2); c) *c2vf* makes use instead of a unique embedding representing the whole sentential context (Section 3.4.2.3).

3.4.2.2 *AddCos*: skip-gram target word and context word embeddings

Melamud et al. (2015)'s method for lexical substitution is based on the word2vec skip-gram word embedding model (Mikolov et al., 2013a). The novelty of Melamud et al.'s approach is that it leverages the word-as-context embeddings in combination with the word-as-target embeddings for the lexical substitution task. The method ranks substitutes based on a measure that combines two types of similarity: a) *target-to-substitute*, showing how similar a potential substitute is to the target word, and b) *target-to-context*, reflecting the substitute's compatibility with a given sentential context. Similarities are estimated using the vector cosine distance between the respective skip-gram embeddings. The measures differ in the way they combine the similarities together, using either an arithmetic or a geometrical mean. Following Apidianaki et al. (2018), we choose the more flexible additive approach which, contrary to the multiplicative variants, does not require high similarities in all elements of the product to highly rank a substitute, but can yield a high score even if one of the elements in the sum is zero. The *Add* measure (Equation 3.4.2.2, hereafter called *AddCos* because of the cosine function used) estimates the substitutability of a candidate substitute s_i of the target word t in context C, where C corresponds to the set of the target word's context elements in the sentence, and c corresponds to an individual context element. In Equation 3.4.2.2, we abuse notation and use t and s_i to refer to the word-as-target *embeddings* of the target word and a possible substitute. c denotes the word-as-context *embedding* of a context word. The amount of context words to be used can be limited to a fixed-size window around the target word. We experiment with $|C| = 2$ and $|C| = 8$ (one and four words at each side of the target, respectively). *cos* refers to the cosine similarity between two vectors.

$$AddCos(t, s_i, C) = \frac{cos(s_i, t) + \sum_{c \in C} cos(s_i, c)}{|C| + 1} \tag{3.1}$$

With this method, we use the 300-dimensional skip-gram word-as-target and word-as-

context embeddings described in Section 3.4.1. We also apply the *AddCos* method to ELMo and BERT, as well as to PSTS-ELMo embeddings. When using standard ELMo/BERT embeddings, the target and context word representations of a sentence (t and c) are their corresponding ELMo/BERT vector in that sentence, and the vector of a candidate substitute s_i is obtained by substituting the target word by the candidate s_i in the same context, as described in Section 3.4.2.1 and Table 3.3. To adapt this to PSTS-ELMo embeddings, substitute representations are replaced by their corresponding PSTS-ELMo vectors.

3.4.2.3 The *context2vec*-based model (*c2vf*)

In the context2vec model, words and contexts are embedded in the same space, which allows for calculating target-to-context, context-to-context and target-to-target similarities. A score for a candidate substitute is computed using the following formula:

$$\text{c2vf}(\text{t}, \text{s}_\text{i}, \text{C}) = \frac{cos(s_i, t) + 1}{2} \times \frac{cos(s_i, C) + 1}{2} \tag{3.2}$$

where t and s_i are the word embeddings of the target and a substitute, and C is the context2vec context vector of the sentence with an empty slot at the target's position.

We also use Equation 3.2 (hereafter called *c2vf*) with standard ELMo and PSTS-ELMo vectors. As with the *AddCos* method, we represent the target word t in context by its ELMo embedding, and the substitute vectors are obtained with the in-place substitution approach described above (cf. Sections 3.4.2.1 and Table 3.3). The context vector (C) is the average of the ELMo embeddings of all words in the context. To test PSTS-ELMo embeddings in this setting, each substitute is represented by its PSTS-ELMo embedding.

We also experiment with PSTS-c2v embeddings, i.e. standard context2vec embeddings tuned on the PSTS dataset. In this configuration, target and context are represented with standard context2vec embeddings, and substitutes are represented with PSTS-c2v embeddings.

Finally, we use context2vec embeddings removing the *target-to-substitute* component of this formula, leaving only *substitute-to-context* similarity (**s2C**). As explained in Section 3.2, the pool of candidate substitutes we use is of high quality, as it contains true paraphrases of target words. We expect target-to-substitute similarity to be less crucial in such conditions.

3.4.2.4 Baselines

We compare our models to a context-insensitive baseline that solely relies on the target-to-substitute similarity of standard pre-trained word embeddings: we use 300-dimensional GloVe vectors (Pennington et al., 2014)[5] and 300-dimensional FastText vectors, both trained on Common Crawl (Mikolov et al., 2018).[6] Similar to *tTs* (Section 3.4.2.1), this approach only considers target-to-substitute similarity, but these static representations do not have any access to context information, and therefore the ranking proposed for a target word is always the same regardless of context.

[5] https://nlp.stanford.edu/projects/glove

[6] https://fasttext.cc/docs/en/english-vectors.html

We also propose an enriched version of the baseline model by creating a simple representation of context: the average of the static embeddings of words in a sentence. We then compare target and substitute vectors to the generated context vector using the context2vec formula (Equation 3.2) We call these models *GloVe + context* and *FastText + context*.

3.5 Evaluation

We compare the performance of the proposed lexical substitution models on the substitute ranking task, where models assign scores to all candidate substitutes in S_t for a target word t according to their suitability in new contexts. For evaluation, we use the test set from the SemEval-2007 Lexical Substitution task. We filter the test set to preserve only target words and substitutes present in PPDB 2.0 (XXL) which have a vector available in all tested models, to ensure all methods use exactly the same substitute pool per target word. Target words for which none or only one substitute was left were removed. The filtered test set used in our experiments includes 168 target words and 1,584 sentences.

The ranking performed by each model is compared to the gold ranking by means of Generalised Average Precision (GAP) (Kishida, 2005). GAP measures the quality of a ranking by comparing the resulting ranked list with the gold standard annotation, using substitution frequency as weights (that is, the number of annotators that suggested each substitute). GAP scores range between 0 and 1. A score of 1 indicates a perfect ranking where all correct substitutes precede all incorrect ones, and high-weight substitutes precede low-weight ones (Thater et al., 2010). We use the GAP implementation from Melamud et al. (2015).[7]

The main formula for GAP is in Equation 3.3. Having n candidate substitutes ranked by a model from most to least suitable, x_i denotes the gold weight (annotation frequency) associated with the ith substitute. y_i refers to the gold weight of the ith substitute in the gold ranking R. $I(x_i)$ (Equation 3.4) is 0 if the ith substitute in the predicted ranking is not present in the gold ranking (i.e. has 0 frequency), and 1 otherwise. $\overline{x}_i$ (Equation 3.5) is the average of the gold weight values up to the ith substitute in the predicted ranking. Analogously, $\overline{y}_i$ is the average of the gold weight values up to the ith substitute in the gold ranking.

$$GAP = \frac{\sum_{i=1}^{n} I(x_i)\overline{x}_i}{\sum_{i=1}^{|R|} I(y_i)\overline{y}_i} \tag{3.3}$$

$$I(x_i) = \begin{cases} 0 & x_i = 0 \\ 1 & x_i > 0 \end{cases} \tag{3.4}$$

$$\overline{x}_i = \frac{\sum_{k=1}^{i} x_k}{i} \tag{3.5}$$

For a clearer picture of the scores assigned by this metric to rankings of different quality, we provide in Table 3.4 a few made-up toy rankings with their corresponding GAP scores. Note

[7] https://github.com/orenmel/lexsub

Gold ranking				
sack (5)	dismiss (1)	shoot (0)	launch (0)	
Example rankings				**GAP**
sack	dismiss	shoot	launch	1.000
sack	shoot	dismiss	launch	0.875
dismiss	sack	shoot	launch	0.500
shoot	dismiss	launch	sack	0.250
launch	shoot	dismiss	sack	0.229

Table 3.4: Examples of GAP scores that would be assigned to made-up example rankings of different quality.

that a GAP score of 0 is obtained when none of the substitutes proposed by the model is present in the gold ranking. This is not possible with our models, as they rank all candidates available, and these come from the LexSub dataset itself. We calculate and report, along with the results, a lower bound of the GAP score in these conditions, corresponding to the GAP of a model that has access to the same pool of substitutes as all our models, but systematically predicts the reverse of the gold ranking.

3.6 Results

The results obtained by the proposed methods in the substitute ranking task are given in Table 3.5. Results show that BERT (*tTs*) outperforms other methods, and context2vec performs better than ELMo in this task. BERT's singularity lies in its training task. Instead of predicting the immediate next word based on the previous (left-to-right) or posterior (right-to-left) token, BERT is trained with a cloze task where words in different parts of the sentence are masked and they have to be predicted using information from the whole sentence. In addition, it has a deeper, Transformer architecture. We think the superiority of context2vec with respect to ELMo is due to its training objective as well: context2vec is explicitly trained with pairs of target words and sentential contexts, optimizing the similarity of context vectors and potential fillers. In contrast, ELMo representations are trained as a general language model that predicts the immediate next tokens, while the target-to-substitute and substitute-to-context similarities used by the lexical substitution methods are not explicitly accounted for. The underlying assumption of the *AddCos* and *c2vf* methods that these similarities need to be high for good substitutes, does not thus apply in the case of ELMo embeddings.

The average and top layer configurations give comparable results both with ELMo and BERT, with the average performing slightly better in all settings. Peters et al. (2018b) present a thorough analysis of the performance of different layers of the biLM models in different tasks, which shows that top layers are better suited for semantic-related tasks or long-distance phenomena than lower layers. In the supervised word sense disambiguation (WSD) evaluation presented in Peters et al. (2018a), results obtained using the top layer were slightly better than

Method	Vectors	GAP	PSTS-Vectors	GAP
AddCos (|C|=2)	Skip-gram (Apidianaki et al., 2018)	0.527		
	ELMo-avg	0.527	PSTS-ELMo-avg	0.494
	ELMo-top	0.513	PSTS-ELMo-top	0.491
AddCos (|C|=8)	Skip-gram (Apidianaki et al., 2018)	0.520		
	ELMo-avg	0.498	PSTS-ELMo-avg	0.481
	ELMo-top	0.476	PSTS-ELMo-top	0.478
c2vf	UkWac c2v (Melamud et al., 2016)	0.587	PSTS-c2v	0.492
	ELMo-avg	0.529	PSTS-ELMo-avg	0.490
	ELMo-top	0.516	PSTS-ELMo-top	0.480
tTs	ELMo-avg (Peters et al., 2018a)	0.534	PSTS-ELMo-avg	0.493
	ELMo-top (Peters et al., 2018a)	0.531	PSTS-ELMo-top	0.488
	BERT-avg (4) (Devlin et al., 2019)	**0.634**		
	BERT-top (Devlin et al., 2019)	**0.627**		
s2C	**c2v** (Melamud et al., 2016)	**0.597**		
Baseline + context (c2vf)	GloVe (Pennington et al., 2014)	0.467		
	Fasttext (Mikolov et al., 2018)	0.491		
Baseline (tTs)	GloVe (Pennington et al., 2014)	0.465		
	Fasttext (Mikolov et al., 2018)	0.485		
GAP lower bound	-	0.156		

Table 3.5: Results of the substitute ranking experiment with all methods and embedding types. For AddCos models, |C| refers to the size of the window: |C|=2 uses one context word at each side of the target.

those of the middle layer. We believe the slight advantage of the *avg* models, compared to *top*, in this task, highlights an important difference between Lexical Substitution and WSD. In Lexical Substitution, the selected substitute needs to correctly describe the meaning of the target word instance *and* to be a good fit in the context, whereas selection in WSD mainly relies on semantic adequacy. For example, when selecting one among available senses of a word in a resource like WordNet, the synonyms found in the selected synset might not all be good in-context substitutes. We believe the ELMo representation obtained by averaging the three layers to contain information regarding the semantic, syntactic and collocational adequacy of a word. This does not contradict previous findings, since the semantic tasks where the top ELMo layer was found to perform best were tasks that involve longer range dependencies and a more general notion of semantic similarity (e.g. coreference resolution).

Aina et al. (2019)'s analysis of word and context information in the hidden representations of a biLSTM language model provides another possible explanation for this outcome with the ELMo model. The representations corresponding to a word t seem to contain more information about t and possible substitutes s_i in the early layers, as the last layers would be more focused on next word prediction.

The results obtained for *PSTS-ELMo-** and *PSTS-c2v* configurations show that ELMo and context2vec representations do not benefit from the addition of substitute-specific data in the form of PSTS sentences, rather the contrary. Whereas it looks like PSTS is introducing

Sentence	*on the **way** out of the parking lot johnny felt a thump*
Candidate substitutes for *way.n*	sense, means, aspect, technique, passage, respect, direction, characteristic, journey, method, route, practice, fashion, manner
Gold ranking	route (3), passage (1), journey (1)

Table 3.6: An instance of the target noun *way* (*way.n*) from the SemEval-2007 test set, its candidate substitutes, and the gold substitute ranking used for evaluation.

Method	**Vectors**	**Ranked substitutes**
c2vf	c2v (Melamud et al., 2016)	**route**, **journey**, manner, **passage**, direction, means, sense, aspect, method, fashion, respect, technique, characteristic, practice
tTs	BERT-avg (4)	**journey**, **route**, manner, **passage**, sense, aspect, direction, method, respect, means, fashion, characteristic, technique, practice
Baseline	GloVe (Pennington et al., 2014)	sense, means, manner, **journey, route,** direction, respect, aspect, practice, method, technique, fashion, **passage**, characteristic
Baseline + ctxt	GloVe (Pennington et al., 2014)	sense, means, manner, direction, respect, **journey**, aspect, **route**, practice, method, **passage**, technique, fashion, characteristic

Table 3.7: Examples of substitute rankings for the instance of the noun "*way*" given in Table 3.6 produced by the two best-performing methods (c2vf with standard c2v embeddings and tTs with *BERT-avg (4)* embeddings) and the two methods with lowest GAP (baseline and baseline + context with GloVe embeddings). Correct substitutes are marked in boldface to highlight their position in the ranking proposed by each model.

confusion to an already good model, we believe this could be due to the small amount of PSTS sentences used for tuning (100), which biases the model towards those sentences. Another reason could be that the top-ranked sentences in PSTS are not always high quality, i.e. they might not contain, or not be representative enough of, the sense being expressed.

The baseline methods, which slightly benefit from the addition of context, are not very far behind most *PSTS-ELMo-** models. FastText vectors are trained with word2vec's CBOW architecture using position-dependent weighting, which results in richer context representations and is, we believe, the main reason of its advantage over GloVe on this task.

The fact that the model which only relies on substitute-to-context similarity (*s2C*) is superior to its counterpart that also uses target-to-substitute similarity (*c2vf, UkWaC c2v*) is probably due to the high quality of the selected pool of substitutes, which come from manual annotations and are therefore correct paraphrases of target words.

Finally, we observe that, for the AddCos method, a smaller context window around the target word (|C|=2) is consistently slightly more effective than a bigger one (|C|=8). This

suggests that the most relevant context clues for lexical substitution are found in the close vicinity of a target word.

In Tables 3.6 and 3.7, we give an example of an instance of the target word *way* and the substitute ranking proposed by some of the models. In Table 3.6, we also provide the candidate substitutes considered for the target word. Numbers in parentheses denote the number of annotators that proposed each substitute. We observe that the stronger models which use the c2v formula with the standard context2vec vectors (trained on UkWac), and the tTs method with BERT-avg (4), rank substitutes better than the baseline models.

To sum up, we have compared different types of representations on the lexical substitution task as a way of evaluating their ability to model word meaning in context. BERT representations, followed by context2vec, are the best representations in this respect.

3.7 Conclusion

We analysed the behavior of different word and context representations in an in-context substitute ranking task. The compared methods differ as to the type of similarity they consider between words (target-to-substitute) and contexts (substitute-to-context). We experimented with the standard representations from each embedding model, and tuning them to the lexical substitution task using an automatically compiled collection of sentences representing target-substitute pairs. Our results show that models trained with a slot-filling objective that optimises the inter-dependencies between candidate substitutes and context, like context2vec and BERT, are a better fit for the Lexical Substitution task than models with more traditional language model objectives focused on next word prediction, like ELMo. This is because they encode target and local context information appropriately for this task, which ensures the semantic and syntactic adequacy of the selected substitutes. BERT and context2vec are thus more suited to representing word meaning in context.

Tuning ELMo and context2vec on the sentences of the PSTS dataset, which represent a specific sense described by a substitute, did not help the models. Still, the resource has potential to be used for lexical substitution in other ways, for example for training supervised neural models for this task.

Recently, a few interesting approaches for Lexical Substitution with BERT and other contextual models were proposed. Zhou et al. (2019) introduced a modification to BERT's architecture which consists in an embedding dropout mechanism that partially masks the target word by setting some of its embedding dimensions to 0. With this procedure, when used as a language model, BERT receives only vague information about the target word and predicts similar –but not identical– words that could replace it. They also propose to rank candidates based on the similarity of the sentence before and after substitution, with the goal of rewarding substitutes that cause a minimal meaning change in the sentence. Arefyev et al. (2020) present an extensive comparison of (masked) language models on the lexical substitution and the Word Sense Induction (WSI) tasks. Similar to Zhou et al. (2019) and contrary to our work where we extract

representations from the models, they exploit the word probabilities assigned by these language models, and experiment with several ways of injecting target word information. They find XLNet (Yang et al., 2019) to perform best on lexical substitution, and obtain state-of-the-art results on a WSI task using the substitutes predicted by this model.

In the upcoming chapter, we continue investigating the quality of word meaning representation in contextual models. We use a task that is highly related to lexical substitution: word usage similarity estimation, and exploit the similarity between the two tasks to improve the models' word usage similarity predictions.

Chapter 4

Word Usage Similarity Estimation

4.1 Introduction

In the previous chapter, we used the lexical substitution task to investigate the ability of different context-sensitive representations to represent word meaning in context. In this chapter, we further explore this question using another task: word usage similarity estimation. This task involves estimating the semantic proximity of word instances in different contexts (Erk et al., 2009). A model that makes good use of the semantic information in a word's context should be able to generate representations that reflect the semantic similarity of word instances. For example, we want the representations of *fan* in the sentences "turn on the *fan*" and "the *fan* is not working" to be similar to each other, and dissimilar from that in "I'm your biggest *fan*".

This task is strongly related to lexical substitution. The set of substitutes proposed by annotators for a word in a sentence represent its meaning. The overlap of substitutes of two instances of the word is an estimate of their semantic proximity: in the first two sentences, *ventilator* would be acceptable, but not *admirer*; whereas *admirer* would be the correct choice in the third sentence.

We present our experiments using word and sentence representations for usage similarity prediction. First, we experiment with an unsupervised approach which relies on the cosine similarity of different kinds of representations. In order to improve the quality of the predictions, we exploit the similarity between word usage similarity and lexical substitution in supervised models. These models combine embedding similarity with features based on substitute overlap.

Usage similarity can be viewed as a classical Semantic Textual Similarity task (Agirre et al., 2012, 2016) with a focus on a particular word in the sentence. This connection motivated us to apply models initially proposed for sentence similarity to usage similarity prediction. We perform an extensive comparison of existing word, context and sentence representation methods on this task, including context2vec, BERT, and the Universal Sentence Encoder (Cer et al., 2018).

Past work (Erk et al., 2009; McCarthy et al., 2016) has used manually proposed substitutes in context as a proxy for measuring the usage similarity of words. We propose to use automatically obtained substitutes, bypassing the need for manual substitute annotations. Automatic

substitutes have proven to be useful for the related task of Word Sense Induction (Alagić et al., 2018). We apply a lexical substitution method from the previous chapter, and use different measures of substitute overlap. We also propose a methodology for collecting new training data for supervised usage similarity estimation from a dataset annotated with lexical substitutes.

We test our models on benchmark datasets containing gold graded and binary word usage similarity judgments: Usim (Erk et al., 2009, 2013) and WiC (Pilehvar and Camacho-Collados, 2019).

A previous attempt at automatic and unsupervised usage similarity prediction involved obtaining vectors encoding a distribution of topics for every target word in context (Lui et al., 2012). Usage similarity was approximated by the cosine similarity of the resulting topic vectors. We show how contextualised representations, and a supervised model that uses them as features, outperform topic-based methods on this task.

We also describe our participation in the SemDeep-5 WiC shared task (Espinosa-Anke et al., 2019), where we applied this methodology to the latest version of the WiC dataset, and present an analysis of BERT's usage similarity estimation capability through all its layers.

Our experiments reveal to what extent the organisation of word instances in the space of different representations reflects their semantics. They also help to determine the utility of substitute-based features for improving word usage similarity predictions.

4.2 Data

The datasets described in this Section have been introduced in more detail in Sections 2.1.3.2 and 2.1.3.3.

LexSub and Usim The SemEval-2007 Lexical Substitution dataset (LexSub) contains instances of words hand-labelled with meaning-preserving substitutes. A subset of LexSub has also been manually annotated with graded pairwise usage similarity judgments (Erk et al., 2009, 2013). The scores range from 1 to 5 (dissimilar/similar word instances). In our experiments, we use 2,466[1] sentence pairs from the Usim dataset for training, development and testing of automatic usage similarity prediction methods.

Table 4.1 shows examples of sentence pairs from the Usim dataset alongside the gold substitutes and usage similarity scores assigned by the annotators. For comparison, we also include in the Table the substitutes selected for these instances by the automatic context2vec substitution method used in our experiments (more details in Section 4.3.2). We also use the gold substitutes in LexSub to train the models, in order to assess the impact of automatic substitutes compared to manual ones on this task.

Concepts-in-Context (CoInCo) Given the small size of the Usim dataset, we extract additional training data for our models from the Concepts in Context (CoInCo) corpus (Kremer

[1]This is the number of pairs that have been assigned a score in Usim for which manual and automatic substitutes are available.

Sentences	Substitutes
The local **papers** took photographs of the footprint.	GOLD: newspaper, journal AUTO-LSCNC: press, newspaper, news, report, picture AUTO-PPDB: newspaper, newsprint
Now Ari Fleischer, in a pitiful letter to the **paper**, tries to cast Milbank as the one getting his facts wrong.	GOLD: newspaper, publication AUTO-LSCNC: press, newspaper, news, article, journal, thesis, periodical, manuscript, document AUTO-PPDB: newspaper
This is also at the very essence or heart of being a **coach**.	GOLD: trainer, tutor, teacher AUTO-LSCNC: teacher, counsellor, trainer, tutor, instructor AUTO-PPDB: trainer, teacher, mentor, coaching
We hopped back onto the **coach** – now for the boulangerie!	GOLD: coach, bus, carriage AUTO-LSCNC: bus, car, carriage, transport AUTO-PPDB: bus, train, wagon, lorry, car, truck, carriage, vehicle

Table 4.1: Examples of highly similar and dissimilar usages from the Usim dataset for the nouns *paper* (Usim score = 4.34) and *coach.n* (Usim score = 1.5), with the substitutes assigned by the annotators (GOLD). For comparison, we include the substitutes that were selected for these instances by the automatic substitution method used in our experiments (based on context2vec embeddings) from two different pools of substitutes (AUTO-LSCNC and AUTO-PPDB). More details on the automatic substitution configurations are given in Section 4.3.2.

et al., 2014), which contains manually selected substitutes for all content words in a sentence. CoInCo provides no usage similarity scores that could be used for training. We construct additional training data as follows: we gather all instances of a target word with at least four substitutes, and keep pairs with (1) no overlap in substitutes, and (2) minimum 75% substitute overlap.[2] We view the first set of pairs as examples of completely different usages of a word (DIFF), and the second set as examples of identical usages (SAME). The two sets are unbalanced in terms of number of instance pairs (19,060 vs. 2,556). We balance them by keeping in DIFF the 2,556 pairs with the highest number of substitutes.

We also annotate the data with substitutes using context2vec (Melamud et al., 2016) (cf. Section 4.3.2). We apply an additional filtering to the sentence pairs extracted from CoInCo, discarding instances of words that are not in the context2vec vocabulary and have no embeddings. We are left with 2,513 pairs in each class (5,026 in total). We use 80% of these pairs (4,018) together with the Usim data to train our supervised Usim models described in Section 4.3.3.[3]

Word-in-Context (WiC) The third dataset we use in our experiments is WiC (Pilehvar and Camacho-Collados, 2019), version 0.1.[4] WiC provides pairs of contextualised target word

[2]Full overlap is rare since annotators propose somewhat different sets of substitutes, even for instances with the same meaning. Full overlap is observed for only 437 of all considered CoInCo pairs (0.3%).

[3]The dataset is available at https://github.com/ainagari/coinco_usim_data/. We kept aside 20% of the extracted examples for development and testing purposes.

[4]More details about this version are found in Pilehvar and Camacho-Collados (2018) and https://pilehvar.github.io/wic/.

instances describing the same or different meaning, framing in-context sense identification as a *binary* classification task. WiC 0.1 comes with an official train/dev/test split containing 7,618, 702 and 1,366 sentence pairs, respectively.

4.3 Methodology

We experiment with two ways of predicting usage similarity. In Section 4.3.1, we present an unsupervised approach that provides direct usage similarity assessments based on the cosine similarity of different kinds of word and sentence representations. We also design a supervised approach that combines embedding similarity with features based on substitute overlap. In Section 4.3.2, we describe how substitute-based features were extracted, and in Section 4.3.3, we introduce the supervised models.

4.3.1 Direct Usage Similarity Prediction

In the unsupervised prediction setting, we apply different types of pre-trained word and sentence embeddings as follows: we compute an embedding for every sentence in the Usim dataset, and calculate the pairwise cosine similarity between the sentences available for a target word. Then, for every embedding type, we measure the correlation between sentence pair similarities and gold usage similarity judgments in the Usim dataset, using Spearman's ρ correlation coefficient. We report the results in Section 4.4. We experiment with the following embedding types:

- **GloVe** embeddings (Pennington et al., 2014): non-contextualised word representations which merge all senses of a word in one vector. We use 300-dimensional GloVe embeddings pre-trained on Common Crawl (840B tokens).[5] The representation of a sentence is obtained by averaging the GloVe embeddings of all words in the sentence.

- **SIF** (Smooth Inverse Frequency) embeddings are sentence representations built by applying dimensionality reduction to a weighted average of static embeddings of words in a sentence (Arora et al., 2017). We use SIF in combination with GloVe vectors.

- **Context2vec** embeddings (Melamud et al., 2016). We use a context2vec model pre-trained on the ukWaC corpus (Baroni et al., 2009)[6] to compute embeddings for sentences with a blank at the target word's position.

- **ELMo** (Peters et al., 2018a). We use a 512-dimensional biLM pre-trained on approximately 800M tokens of news crawl data.[7] We use out-of-the-box embeddings (without tuning) and experiment with the top layer, and with the average of the three hidden layers. We represent a sentence in two ways: with the contextualised ELMo embedding

[5] https://nlp.stanford.edu/projects/glove/
[6] https://github.com/orenmel/context2vec
[7] https://allennlp.org/elmo

obtained for the target word, and with the average of ELMo embeddings for all words in a sentence.

- **BERT** (Devlin et al., 2019). We use the average of the last 4 layers of the `bert-base-uncased`[8] model and create target word and sentence representations in the same way as for ELMo: using either the BERT embedding of the target word,[9] or the average of the BERT embeddings for all tokens in a sentence.

- **Universal Sentence Encoder (USE)** (Cer et al., 2018) makes use of a Deep Averaging Network (DAN) encoder that averages word and bigram embeddings and passes them to a feedforward network to create sentence representations. The model is trained in a multitask setting and has been shown to improve performance on different NLP tasks through transfer learning.[10]

- **doc2vec** is an extension of word2vec to the sentence, paragraph or document level (Le and Mikolov, 2014). One of its forms, dbow (distributed bag of words), is based on the skip-gram model, where it adds a new feature vector representing a document. We use a dbow model trained on English Wikipedia released by Lau and Baldwin (2016).[11]

4.3.2 Substitute-based Feature Extraction

In this Section we present our methodology for ranking substitutes for word instances (Section 4.3.2.1), and for selecting the higher-ranked substitutes, which best describe the meaning of each instance (Section 4.3.2.2). We use these substitutes to extract features for our supervised word usage similarity models (Section 4.3.2.3).

4.3.2.1 Automatic Lexical Substitution

We generate rankings of substitutes for words in context using the context2vec-based method with context2vec embeddings (Melamud et al., 2016). This method has been described in the Lexical Substitution Chapter (Section 3.4.2.3). It performs well and is not as computationally expensive as the BERT-based lexical substitution model (Section 3.4.2.1). We use two pools of candidates: (a) paraphrases of the word in the Paraphrase Database (PPDB) 2.0 XXL package (Ganitkevitch et al., 2013; Pavlick et al., 2015) (AUTO-PPDB), and (b) substitutes that were proposed for each word in LexSub and CoInCo (AUTO-LSCNC). In our experiments on the WiC dataset, where no substitute annotations are available, we only use AUTO-PPDB as our candidate pool. We use the *s2C* method described in Section 3.4.2.3 (which relies on substitute-to-context similarity only) for AUTO-LSCNC, because substitutes have been manually selected and are, therefore, of high quality. Substitutes are semantically similar to the target, consequently context2vec just needs to rank them according to how well they fit the new context. For

[8] https://github.com/google-research/bert

[9] When a word is split into multiple word pieces (Wu et al., 2016), we average them to obtain its representation.

[10] https://tfhub.dev/google/universal-sentence-encoder/2

[11] https://github.com/jhlau/doc2vec

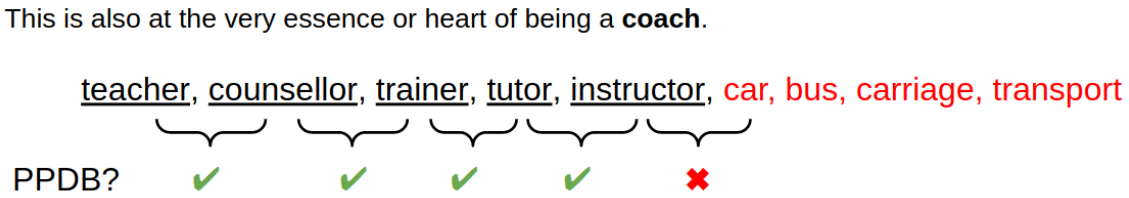


Figure 4.1: The PPDB filtering strategy finds a cut-off point in a substitute ranking by checking what adjacent substitutes are not a paraphrase pair in PPDB. The absence of a pair in PPDB is seen as a change in meaning in the ranking.

AUTO-PPDB, we instead use the full *c2vf* formula. We do this because PPDB can contain noisy candidates that are not good paraphrases, due to it being built automatically, and target-to-substitute similarity can help rank them lower.

Following this procedure, we obtain a ranking of the candidate substitutes for each word instance in the Usim, CoInCo and WiC datasets.

4.3.2.2 Substitute Filtering

For every target word instance, all candidate substitutes available for the target in each pool are ranked. Consequently, the automatic annotations produced for different instances of the target all include the same set of substitutes, but in different order. This does not allow for the use of measures based on substitute overlap. In order to use this type of measures, we propose ways to filter the generated rankings, and keep for each instance only substitutes that are a good fit in context. We test different filters to discard low quality substitutes from the annotations proposed by context2vec for each instance:

- **PPDB**: Given a ranking R of n substitutes $R = [s_1, s_2, ..., s_n]$ proposed by context2vec, we form pairs of substitutes in adjacent positions $\{s_i \leftrightarrow s_{i+1}\}$, and check whether they exist as paraphrase pairs in PPDB 2.0 XXL. We expect substitutes that are paraphrases of each other to be similarly ranked. If s_i and s_{i+1} are not paraphrases in PPDB, we keep all substitutes up to s_i and use this as a cut-off point, discarding substitutes present from position s_{i+1} onwards in the ranking. The idea is that good quality substitutes should be both high-ranked and semantically related. Figure 4.1 illustrates the process followed in this filtering strategy along with an example from the Usim dataset.
- **GloVe word embeddings**: We measure the cosine similarity (*cos*) between GloVe embeddings of adjacent substitutes $\{s_i \leftrightarrow s_{i+1}\}$ in the ranking R obtained for a new instance. We first compare the similarity of the first pair of substitutes ($cos(s_1, s_2)$) to a lower bound similarity threshold T. If $cos(s_1, s_2)$ exceeds T, we assume that s_1 and s_2 have the same meaning, and use $cos(s_1, s_2)$ as a reference similarity value, S, for this instance. The middle point between the two values, $M = (T + S)/2$, is then used as a threshold to determine whether there is a shift in meaning in subsequent pairs. If $cos(s_i, s_{i+1}) < M$, for $i > 1$, then only the higher ranked substitute (s_i) is retained and all subsequent substitutes in the ranking are discarded. The intuition behind this calculation is that if

cos is much lower than the reference S (even if it exceeds T), substitutes possibly have different senses.

- **Highest-ranked X substitutes**. We also test two simple baselines, which consist in keeping the 5 and 10 highest-ranked substitutes for each instance.

We test the efficiency of each filter on the portion of the LexSub dataset that was not annotated for Usage Similarity. We compare the substitutes retained for each instance after filtering to its gold LexSub substitutes. Filtering results are reported in Appendix A.1.1.

The best filters were GloVe word embeddings ($T = 0.2$) for AUTO-LSCNC, and the PPDB filter for AUTO-PPDB.

4.3.2.3 Feature Extraction

After annotating the Usim sentences with substitutes and filtering, we extract features related to the extent of substitute overlap. For each sentence pair with rankings R_1 and R_2, we obtain the following features.

- **Common substitutes**. The proportion of shared substitutes between the two instances of a target word, as shown in equation 4.1

$$\text{common substitutes} = \frac{|R_1 \cap R_2|}{|R_1 \cup R_2|} \tag{4.1}$$

- **GAP score**. The average of the Generalised Average Precision (GAP) score (Kishida, 2005) taken in both directions ($GAP(R_1, R_2)$ and $GAP(R_2, R_1)$). GAP (introduced in detail in Section 3.5) is a measure that compares two rankings considering not only the order of the ranked elements but also their weights. We use the frequency in the manual LexSub annotations (i.e. the number of annotators who proposed each substitute) as the weight for gold substitutes, and the context2vec score for automatic substitutes.
- **Substitute cosine similarity**. We form substitute pairs ($R_1 \leftrightarrow R_2$) and calculate the average of their GloVe cosine similarities. This feature shows the semantic similarity of substitutes, even when overlap is low.

4.3.3 Supervised Usim Prediction

We train linear regression models to predict Usim scores for word instances in different contexts using as features the cosine similarity of the different representations (from Section 4.3.1), and the substitute-based features described in 4.3.2.

In order to be able to evaluate the performance of our models separately for each of the 56 target words in the Usim dataset, we train a separate linear regression model for each word in a leave-one-out setting. Each time, we use 2,196 pairs for training, 225 for development and 45[12] for testing. Each model is evaluated on the sentences corresponding to the target word that was

[12]With the exception of four lemmas which had 36 pairs, and one which had 44.

	Embeddings	Correlation
Full sentence embedding	GloVe	0.142
	SIF	0.274
	c2v	**0.290**
	USE	0.272
	doc2vec	0.124
	ELMo avg	0.254
	ELMo top	0.248
	BERT avg (4)	0.289
Target word embedding	ELMo avg	0.166
	ELMo top	0.177
	BERT top	0.514
	BERT avg (4)	**0.518**

Table 4.2: **Direct usage similarity prediction results:** Spearman's ρ correlations of sentence and word instance embeddings on the Usim dataset. For BERT and ELMo, *top* refers to the top layer, and *avg* denotes the average of layers (3 for ELMo and the last 4 for BERT).

left out. We report results of these experiments in Section 4.4. We compare the performance of the model with context2vec substitutes from the two substitute pools to that of the model with gold substitute annotations. We repeat the experiments by adding CoInCo data to the Usim training data and observing the effect of this additional training data on the results.

To test the contribution of each feature, we perform an ablation study on the 225 Usim sentence pairs in the development set, which cover the full spectrum of Usim scores (from 1 to 5). We report results of the feature ablation in Appendix A.1.2.

We also build a model for the binary Usage Similarity task on the WiC 0.1 dataset, using the official train/dev/test split. We train a logistic regression classifier on the training set, and use the development set to select the best among several feature combinations. We report results of the best performing models on the WiC test set in Section 4.4. For word instances in WiC where no PPDB substitutes are available,[13]we back off to a model that only relies on the embedding features.

4.4 Results

Direct Usim Prediction Correlation results between Usim judgments and the cosine similarity of the embedding representations described in Section 4.3.1 are found in Table 4.2. We observe that target word BERT embeddings give the best performance in this task ($\rho = 0.518$). Context2vec sentence representations are the next best performing representation after BERT, but their correlation is much lower ($\rho = 0.290$). The simple GloVe-based SIF approach for sentence representation, which consists in applying dimensionality reduction to a weighted

[13]2.4%, 2.8% and 9.7% of instances in the training, development and test sets, respectively.

Training set	Features	Gold	AUTO-LSCNC	AUTO-PPDB
Usim	Substitute-based	0.563	0.273	0.148
	Embedding-based	0.494	0.494	0.494
	Combined	**0.626**	0.501	0.493
Usim + CoInCo	Substitute-based	-	0.262	0.129
	Embedding-based	-	0.495	0.495
	Combined	-	0.501	0.491

Table 4.3: **Graded usage similarity results**: Spearman's ρ correlation results between supervised model predictions and graded annotations, averaged by target word. The first column reports results obtained using gold substitute annotations for each target word instance. The last two columns give results with automatic substitutes selected among all substitutes proposed for a word in the LexSub and CoInCo datasets (AUTO-LSCNC), or paraphrases in the PPDB 2.0 XXL package (AUTO-PPDB). The Embedding-based configuration uses cosine similarities from BERT and context2vec, and the Combined configuration includes both kinds of features.

average of GloVe vectors of the words in a sentence, is much superior to the simple average of GloVe vectors and even better than doc2vec sentence representations (which obtain the worst results), and are on par with the more complex USE model. ELMo embeddings work better at the sentence level than at the target level, while the opposite is true for BERT.

Graded Usage Similarity To evaluate the performance of our supervised models, we again measure the correlation of the predictions with human similarity judgments on the Usim dataset using Spearman's ρ. Results reported in Table 4.3 are the average of the correlations obtained for each target word with gold and automatic substitutes from the two substitute pools. It also contains results for each type of features, substitute-based and embedding-based (cosine similarities from BERT and context2vec, the two best performing types of embedding). We also report results with the additional CoInCo training data (Usim + CoInCo). Unsurprisingly, the best results are obtained by the methods that use the gold substitutes. This is consistent with previous analyses by Erk et al. (2009) who found overlap in manually-proposed substitutes to correlate with Usim judgments. The lower performance of features that rely on automatically selected substitutes (AUTO-LSCNC and AUTO-PPDB) demonstrates the impact of substitute quality on the contribution of this type of features. Performance is lowest when candidate substitutes come from an automatic resource (AUTO-PPDB). The addition of CoInCo data does not seem to help the models, especially when substitute-based features are used. This could be due to the fact that CoInCo data contains only extreme cases of similarity (SAME/DIFF) and no intermediate ratings. The slight improvement in the combined settings over embedding-based models is not significant in AUTO-LSCNC substitutes, contrary to when gold substitutes are used ($p < 0.001$).[14]

For comparison to the topic-modelling approach of Lui et al. (2012), we also evaluate on the 34 lemmas used in their experiments. They report a correlation calculated over all instances. With the exception of the substitute-only setting with PPDB candidates, all of our Usim models

[14]As determined by paired t-tests, after verifying the normality of the differences with the Shapiro-Wilk test.

Training set	Features	Accuracy
WiC	Embedding-based	63.62
	Combined	**64.86**
	DeConf embeddings (Pilehvar and Camacho-Collados, 2018)	59.4
	Random baseline (Pilehvar and Camacho-Collados, 2018)	50.0
WiC + CoInCo	Embedding-based	63.69
	Combined	64.42

Table 4.4: **Binary usage similarity results**: Accuracy of models on the WiC 0.1 test set. The Embedding-based configuration includes cosine similarities of BERT avg (4) and USE. The Combined setting uses, in addition, substitute overlap features (AUTO-PPDB).

get higher correlation than their model ($\rho = 0.202$), with $\rho = 0.512$ for the combination of AUTO-LSCNC substitutes and embeddings. The average of the per target word correlation in Lui et al. (2012) ($\rho = 0.388$) is lower than that of our AUTO-LSCNC model in the combined setting ($\rho = 0.500$).

Binary Usage Similarity We evaluate the predictions of our binary classifiers by measuring accuracy on the test portion of the WiC dataset. Results for the best configurations for each training set (WiC and WiC + CoInCo) are reported in Table 4.4. Experiments on the development set showed that target word BERT representations and USE sentence embeddings are the best-suited for WiC. Therefore, 'embedding-based features' here refers to these two representations. Results on the development set can be found in Appendix A.1.3. All configurations obtain higher accuracy than the previous best reported result on this dataset (59.4), obtained using DeConf vectors, which are multi-prototype embeddings based on WordNet knowledge (Pilehvar and Collier, 2016). Adding substitute-based features to embedding features (the Combined setting), despite using the lower-quality AUTO-PPDB substitute pool, slightly improves the accuracy of the model. Also, combining the CoInCo and WiC data for training does not have a clear impact on results, even in this binary classification setting.

4.5 Discussion

We have reported results for the whole Usim dataset, but the strength of the correlation varies greatly for different words in all models and settings. For example, in the case of direct usage similarity predictions with embeddings using BERT, Spearman's ρ ranges from 0.805 (for the verb *fire*) to -0.111 (for the verb *suffer*). This variation in performance is not surprising, since annotators themselves found some lemmas harder to annotate than others, as reflected in the Usim inter-annotator agreement measure (Uiaa) (McCarthy et al., 2016). We find that BERT embedding results correlate with Uiaa per target word ($\rho = 0.59, p < 0.05$), showing that the performance of this model depends to a certain extent on the ease of annotation for each lemma. Uiaa also correlates with the standard deviation of average Usim scores by target word ($\rho = 0.66, p < 0.001$). For example, average Usim values for the word *suffer* do not exhibit high variance as they only range from 3.6 to 4.9. Within a smaller range of scores,

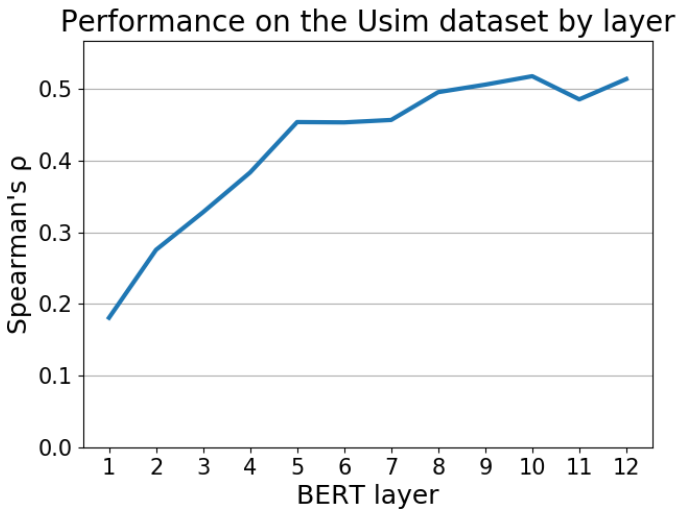


Figure 4.2: Spearman's ρ coefficient obtained with target word instance representations from every layer of the `bert-base-uncased` model.

it is harder to obtain a strong correlation. We also find a negative correlation between Uiaa and the proportion of mid-range judgments for a lemma, a measure called Umid (McCarthy et al., 2016) (-0.46, $p < 0.001$). This also suggests that words with higher disagreement tend to exhibit a higher proportion of mid-range judgments, and fewer extreme (1 or 5) judgments. This analysis highlights the difference between usage similarity across target words.

Interpretability work (Rogers et al., 2020) explores the knowledge that is encoded in deep language models, often trying to pinpoint specific layers or attention heads that contain certain kinds of linguistic information (Tenney et al., 2019a; Voita et al., 2019b). Inspired by this line of work, we evaluated the representations at each layer of the `bert-base-uncased` model on the usage similarity task. Figure 4.2 shows the correlation obtained with each layer on the Usim dataset. We observe an almost steady increase in performance through layers, with a peak at the 10th layer. This layer reaches a $\rho = 0.518$, the same result as the average of the last 4 layers used in our experiments (cf. Table 4.2).

4.6 Exploring Different Context Windows

We also tested the representation types that were used for direct usage similarity prediction using a smaller *context window* (cw) around the target word. Sentences in the WiC dataset are quite short (7.9 ± 3.9 words), but the length of sentences in the Usim and CoInCo datasets varies a lot (27.4 ± 13.2 and 18.8 ± 10.2, respectively). We want to check whether information surrounding the target word in a sentence is more relevant and sufficient for usage similarity estimation. We hypothesise that in long sentences, words situated at a longer distance of the target word may tend to introduce information that is not relevant to the task. We focus on the words in a context window of $\pm$ 2, 3, 4 or 5 words at each side of a target word. Then, we average the word embeddings in this window (for GloVe, ELMo and BERT). We also experiment with excluding the target word instance representation.

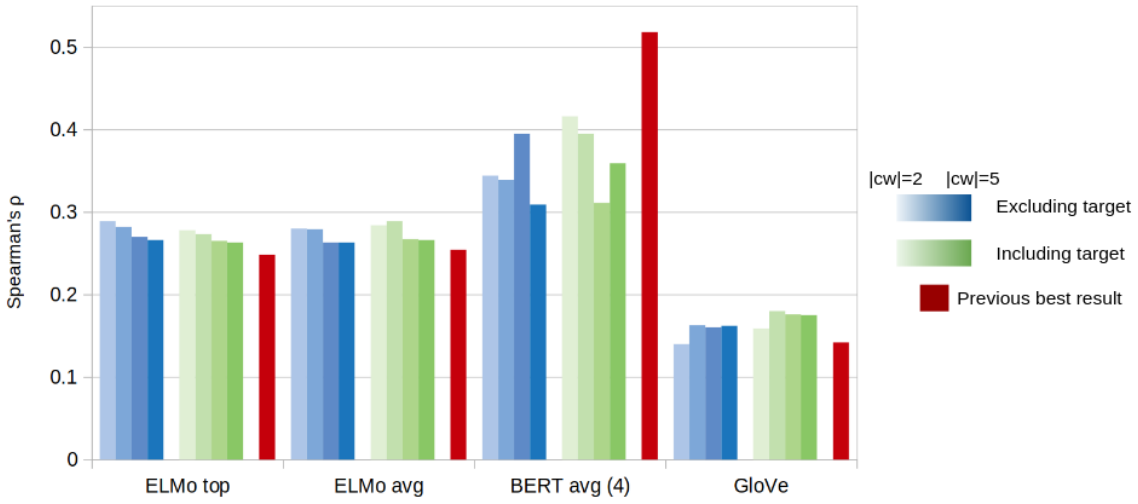


Figure 4.3: Correlation between Usim annotations and cosines of representations obtained from context windows of different sizes. Blue columns indicate contexts *excluding* the representation of the target word, and green columns show results *including* the target word. A darker colour indicates more context words used in the window, from 2 to 5. The red column for each embedding type represents the best result reported earlier (in Table 4.2).

Results of these experiments are found in Figure 4.3. Selecting a context window around (or including) the target word results in worse performance for BERT, for which the target representations gave the best results. It is, however, beneficial for ELMo and GloVe. For these models, using words in a context window is more effective than using words from the whole sentence. The number of words that yields best performance is different depending on the model. For ELMo, the smallest window (|cw|=2) works best, probably because during bidirectional language model training, the words immediately preceding and following the target are used for target word prediction. ELMo is the only model where excluding the target word is better than including it. With GloVe, the best results are obtained with |cw|= 3.

4.7 Participation in the SemDeep-5 WiC Shared Task

Shortly after developing the models described in this chapter, the SemDeep-5 WiC shared task (Espinosa-Anke et al., 2019) was announced. Seven teams proposed models for binary usage similarity in context which were evaluated on WiC version 1.0. This version of WiC contains 7,466 sentence pairs. We participated in the task with the supervised model (Section 4.3.3) for an additional evaluation.

4.7.1 Model Development

We train logistic regression classifiers on the WiC training set and experiment with different feature combinations on the development set. We use cosine similarities from different embedding representations. We exclude GloVe and doc2vec representations from this evaluation because of their low performance on the Usim dataset. For ELMo, we apply a context window

Target		Sentences	Substitutes
T	way	Do you know the **way** to the airport?	ways, route, path, road {*connection, means, journey, move, direction, gateway, passage, place, ...*}
		He said he was looking for the **way** out.	ways, path, road, route, walk {*day, right, passage, move, means, time, doorway, ...*}
F	drink	Can I buy you a **drink**?	beer {*bottle, beverage, pint, vodka, booze, whisky, wine, liquor, drunk, cocktail, restaurant, ...*}
		He took a **drink** of his beer and smacked his lips.	swig {*bottle, pint, sip, drinking, beverage, drank, beer, drunk, cup, booze, liquor, ...*}

Table 4.5: Sentence pairs from the WiC training set for the noun *way* (gold label: T) and the verb *drink* (gold label: F) with automatic substitute annotations assigned by context2vec. Substitutes in italics were discarded after filtering.

of size 2 (not including the target word), since this was the configuration that obtained the best results with ELMo in the Usim experiments (with the top layer, cf. Section 4.6). For BERT, we used the target word representation, averaged across the last four layers. We annotated the dataset with automatic substitutes from the AUTO-PPDB pool. Table 4.5 contains examples of WiC 1.0 instances with substitutes proposed by context2vec and filtered with the PPDB filtering strategy (Section 4.3.2.2). We combine up to four of the best embedding features and train models with the substitute-based features only, backing off to the best embedding-based model for words not present in PPDB.[15] We combine the best embedding- and substitute-based features in the Combined setting. We repeat the experiments with the additional CoInCo training data.

Results on the WiC development set are given in Table 4.6. The best result is obtained by the model trained only on WiC that uses cosine similarities from BERT, USE and ELMo. In the WiC+CoInCo setting, the Combined model gets the same performance as the model that uses the four best embedding types (BERT, USE, ELMo and c2v). We apply the simpler embedding-based model to the WiC test set.

4.7.2 Results and Analysis

Results of the two best-performing models (in boldface in Table 4.6) on the WiC test set are given in Table 4.7. Our best model is the one trained only on WiC, which uses BERT, USE and ELMo cosine similarities. It was ranked third at the competition with an accuracy of 66.71, which is higher than all results reported in the WiC description paper (Pilehvar and Camacho-Collados, 2019).

The additional training data extracted from CoInCo do not help the models. We believe this to be due to the different kind of sense distinctions present in the dataset extracted from CoInCo, and in WiC. To explore this hypothesis, we take a closer look at the predictions of the two best models on the development set and carry out a qualitative analysis of the sense distinctions in

[15] In this version of WiC, 5% of sentence pairs contain target words that are not present in the PPDB XXL package.

Features	WiC	WiC+CoInCo
BERT avg 4 tw	66.46	65.99
USE	63.64	63.48
ELMo top $\|cw\| = 2$	62.38	61.76
SIF	60.66	59.56
c2v	60.34	61.13
BERT, USE	67.87	68.03
BERT, USE, ELMo	**68.65**	68.18
BERT, USE, ELMo, SIF	68.03	-
BERT, USE, ELMo, c2v	-	**68.34**
Substitute-based	60.34	57.84
Combined	66.77	68.34

Table 4.6: Accuracy of the models with embedding-based and substitute-based features on the WiC development set. We report results of the models trained only on WiC, and on the extended (WiC+CoInCo) dataset. We apply the best configurations (marked in boldface) to the WiC test set.

Approach	Accuracy
WiC: BERT, USE, ELMo	**66.71**
WiC+CoInCo: BERT, USE, ELMo, c2v	65.64
BERT_{large} Threshold (Pilehvar and Camacho-Collados, 2019)	63.8

Table 4.7: Accuracy of our two best models on the WiC 1.0 test set, compared to the best result from previous work.

the two datasets. The confusion matrices of the two best models on the development set show that wrong predictions most often concern dissimilar (F) sentence pairs. This type of error occurs more often with the model trained on WiC+CoInCo (67% of total errors compared to 59% when training only on WiC). A quick observation of WiC data reveals that dissimilar (F) pairs sometimes describe related senses, in spite of the pruning that aimed at excluding these from the dataset (Pilehvar and Camacho-Collados, 2019).

We extract a random sample of 120 sentence pairs, 60 from the CoInCo training data and 60 from the WiC development set to explore whether they differ in this respect. We manually annotate all pairs for graded usage similarity, using a scale of 1 (completely different) to 5 (the same), as in Erk et al. (2009). Our assumption is that F pairs that describe related senses will be assigned higher similarity scores. A comparison of the graded usage similarity values of gold F instances reveals that these values differ significantly in CoInCo and WiC ($p = 0.048$), as determined by a Mann-Whitney test, with WiC F pairs having a higher average similarity score (3.19 ± 1.52) than CoInCo F pairs (2.53 ± 0.19). The following F sentence pair from WiC is an example where the target word (*construction*) expresses different but closely related meanings (as a process and as a result):

(1) *Construction* is underway on the new bridge

(2) The engineer marvelled at his *construction*.

The CoInCo sentence pairs that we use for training describe more clear-cut sense distinctions due to the process used for their extraction, which is based on the overlap of manually annotated substitutes.

4.8 Conclusion

We explored the ability of word and context representations to encode the meaning of words in context through the usage similarity estimation task. The task consists in comparing the meaning of two word instances without using word senses from external inventories. We applied a wide range of existing representations to graded and binary usage similarity prediction. In order to improve predictions, we also proposed supervised models that combine similarities from embeddings with features based on lexical substitutes, which describe the meaning of words in context.

Our results show that BERT's semantic space reflects human similarity judgments more accurately than the other representations tested. We also found that the upper layers of the model contain the information most relevant to the task. Another important takeaway is that although substitute annotations are helpful for prediction in supervised models, their quality has a strong impact on performance.

We also observed that usage similarity prediction is much harder for some lemmas than others. This is because of differences in the type of ambiguity: it is generally easier to make predictions for lemmas with clear-cut sense distinctions (like *fire*) than for others with fuzzy distinctions (such as *suffer*). McCarthy et al. (2016) propose methodology for usage similarity estimation with the goal of estimating the ease of partitioning a word into senses. In the following chapter, we also focus on usage similarity on a per lemma basis, trying to identify ambiguous and vague lemmas using contextualised representations.

Chapter 5

Word Sense Clusterability Estimation

5.1 Introduction

In Chapter 2, we described the challenging question of how to establish boundaries between word senses. Polysemous words can have distinct or inter-related meanings, determined to different extent by the context of use (Tuggy, 1993). For example, it is easy to distinguish the MUSIC and STONE senses of the ambiguous noun *rock*, but the meanings of the word *thing* are harder to tell apart; it can refer to different objects in the world or the discourse, and its usages might be more or less related. A polysemous word like *man* would lie somewhere in the middle in the continuum between ambiguity and vagueness, as its different senses (ADULT MALE PERSON, HUMAN, SOLDIER, etc.) are highly related.

McCarthy et al. (2016) propose a method for automatically situating lemmas on a spectrum from ambiguity to vagueness according to their *partitionability*, that is, "the ease with which their usages can be grouped into senses". For example, the instances of the ambiguous word *rock* are easier to group into senses than those of the noun *thing* which has vague semantics. They estimate the partitionability of a lemma in terms of the *clusterability* of its instance representations.

Clusterability measures the extent to which a dataset has a clustered structure, or how easy it is to obtain a meaningful partition of the data (Ackerman and Ben-David, 2009), and thus helps decide whether it is appropriate to proceed with a clustering analysis for a given dataset. McCarthy et al. (2016) create vector representations for word instances from manual substitute and translation annotations, as these approximate the meaning of words in context, and use existing clusterability metrics on these representations to determine the partitionability of a lemma. The need for manual annotations, however, constrains the method's applicability to specific datasets.

In this chapter, we continue our investigation of the quality of the semantic space built by different contextualised representations by evaluating their ability to estimate words' clusterability level. We propose to extend and scale up McCarthy et al.'s work representing word

instances with contextualised representations (Melamud et al., 2016; Peters et al., 2018a; Devlin et al., 2019) and automatically obtained substitutes. Following McCarthy et al. (2016), we cluster word instances using the proposed representations, and apply a set of clusterability metrics to test their partitionability into senses. We also propose to use automatic usage similarity estimations directly (as in Section 4.3.1) for clusterability prediction. These reflect the proximity between word instances in the vector space (Chapter 4), and this information can be used to calculate their clusterability. As in past work, we use partitionability estimates derived from the Usim dataset (Erk et al., 2009, 2013) for evaluation. In concurrent work exploring BERT's semantic space, Yenicelik et al. (2020) also calculate the clusterability of the representations of polysemous words, but do not investigate whether the estimations correlate with partitionability.

Knowing the clusterability of a lemma has several possible applications. Clusterability estimations can help lexicographers determine the number of entries for a word to be present in a resource, and plan the time and effort needed in semantic annotation tasks (McCarthy et al., 2016). They could also guide cross-lingual transfer, serving to identify less clusterable words for which transfer may be harder.

Importantly, clusterability information can help determine whether explicitly modelling the different senses of a lemma would result in meaningful representations, or if it is preferable to process individual instances of a word in context. In other words, it can help select the optimal computational representation for different words. We have presented different types of word representations (Section 2.2), at the type level (static representations (Mikolov et al., 2013a)), sense and multi-prototype embeddings (Reisinger and Mooney, 2010; Neelakantan et al., 2014; Iacobacci et al., 2015) and contextualised vectors (Peters et al., 2018a; Devlin et al., 2019). A per-sense approach might be preferable for words with clear-cut sense distinctions, whereas an instance-per-instance approach, where meaning is dynamically defined by the context of use, could be a better solution for words with vague semantics. Previous studies exploring the question of sense representation adopt a uniform approach (either clustering contexts, or modelling individual instances) without accounting for the properties of a word's semantic space. In this chapter, we explore this idea further. We investigate whether having different types of representations for clusterable and non-clusterable words is beneficial for semantic tasks. Specifically, we propose to modify BERT instance representations of clusterable words, converting them into multi-prototype representations. In another concurrent study, Chronis and Erk (2020) also turn contextualised representations into multi-prototype ones, but they do it for all words in their experiments, regardless of their clusterability level. They find this approach beneficial on out-of-context similarity and relatedness tasks. We, instead, evaluate this approach on the WiC dataset, where models must determine whether two word instances are used in the same sense. For this experiment, we use clusterability estimations that we obtain automatically for a large vocabulary.

Our experiments allow us to learn more about the quality of different types of contextualised representations, and provide interesting insight regarding the feasibility of scaling up clusterability predictions to unrestricted text.

5.2 Methodology

In this section we describe the methodology that we propose for word sense clusterability estimation and how it differs from the approach of McCarthy et al. (2016). In Section 5.2.1, we present the kinds of embeddings that we use to represent Usim word instances in context. In Section 5.2.2 we discuss clusterability estimation in detail, including the initial clustering step (Section 5.2.2.1) and the clusterability metrics used (Section 5.2.2.2).

5.2.1 Word Usage Representations

We represent target word instances in the Usim dataset (Erk et al., 2009, 2013) in two ways: using **contextualised representations** and **substitute-based representations** with automatically generated substitutes. The substitute-based approach allows for a direct comparison with the method of McCarthy et al. (2016).

Contextualised representations We use BERT (Devlin et al., 2019), ELMo (Peters et al., 2018a) and context2vec (Melamud et al., 2016) to generate representations for word instances in Usim. We obtain contextualised ELMo embeddings for instances of a target word w using the second and third layer[1] from the ELMo 1024-d 5.5B model. We generate BERT representations from every layer of the `bert-base-uncased` 768-d model. When a word is split into multiple word pieces, we average them to obtain its representation. We also generate an embedding for the context of each instance using a 600-d context2vec model pre-trained on the UkWac corpus (Baroni et al., 2009).

As shown in Chapter 4, BERT representations give promising results in the related task of usage similarity, showing they successfully capture word meaning in context. For this reason, we also experiment with clustering based on the cosine distance matrix obtained with BERT representations. More details about the different clustering approaches used in our experiments are found in Section 5.2.2.1.

Substitute-based representations Additionally, we represent instances using a substitute-based method, similar to that of McCarthy et al. (2016), but using automatic substitutes instead of manual annotations. We use two different methods for automatic substitution: the context2vec-based method (*c2vf*, introduced in Section 3.4.2.3) using context2vec embeddings, and the *tTs* method (Section 3.4.2.1) with the average of the last four layers in BERT.

We generate our substitutes for each instance i of a target word t in Usim using as candidates S_t the paraphrases of t in the Paraphrase Database (PPDB) XXL package (Ganitkevitch et al., 2013; Pavlick et al., 2015) . For each instance i of t, we obtain a ranking R of all substitutes in S_t. We remove low-quality substitutes (i.e. noisy paraphrases or substitutes referring to a different sense of t) by using the PPDB filtering approach proposed in Section 4.3.2.2. Specifically, we check for each pair of substitutes in subsequent positions in R, starting from the top, whether

[1]We do not use the first layer of ELMo individually. It is character-based, so most representations of a lemma are identical and we cannot obtain meaningful clusters.

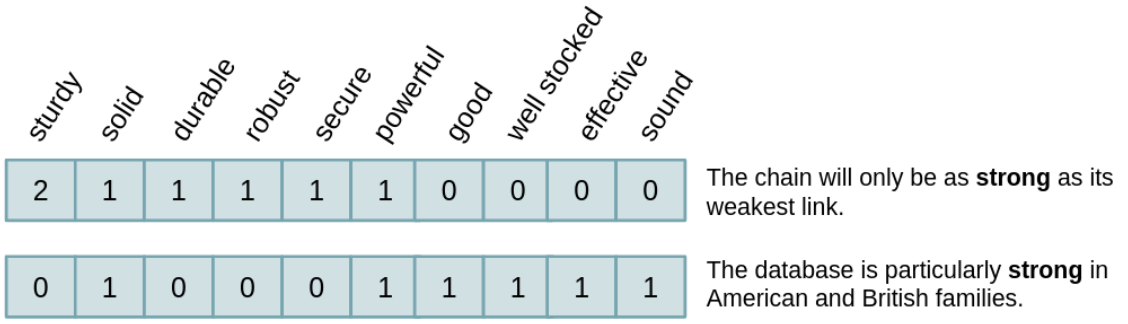


Figure 5.1: Illustration of Manual-SUB representations for instances of the adjective *strong* in the LexSub dataset (McCarthy and Navigli, 2007).

they are paraphrases of each other. If a pair is unrelated in PPDB, all substitutes from that position onwards are discarded.

McCarthy et al. (2016) represent each instance i of a word t in Usim as a vector $\vec{i}$, where each substitute s assigned to t over all its instances $i \in I_t$ becomes a dimension (d_s). For a given i, the value for each d_s in $\vec{i}$ is the number of annotators who proposed substitute s. d_s contains a zero entry if s was not proposed for i. We refer to this type of representation as Manual-SUB, and provide an illustration of how -SUB vectors are built in Figure 5.1.

We build vectors as in McCarthy et al. (2016), using the scores assigned by the lexical substitution methods as a value for each dimension d_s. We call these representations c2v-SUB and BERT-SUB. We also propose an alternative type of representation (c2v-SUBVECS and BERT-SUBVECS) where we average the c2v/BERT (avg (4)) embeddings of the substitutes retained after filtering for each instance $i \in I_t$.

5.2.2 Clustering and Clusterability

Ackerman and Ben-David (2009) and McCarthy et al. (2016) use clusterability metrics initially proposed for estimating the quality of the optimal clustering that can be obtained from a dataset; the better the quality of this clustering, the higher the clusterability of the dataset it is derived from (Ackerman and Ben-David, 2009). We use the same metrics as McCarthy et al. (2016), which require a preliminary clustering step, described in Section 5.2.2.1. We additionally try a clusterability metric that is independent of any clustering algorithm, the Dip's test. Our clusterability metrics are described in detail in Section 5.2.2.2.

5.2.2.1 Determining the number of clusters

We group the word instance representations using k-**means**, as in McCarthy et al. (2016). This clustering algorithm requires the number of clusters (or senses) for a lemma to be specified in advance. In our work this is determined separately for every lemma, without recourse to external resources. McCarthy et al. (2016) use a graph-based approach for determining the number of senses, where word instances are linked by an edge (and belong to the same cluster) based on the overlap of their substitutes. We do not use this method in our experiments,

because it is not compatible with contextualised representations, and it also requires defining a distance threshold.

To define the optimal number of senses (k) for a specific lemma, we instead perform k-means clustering for a range of k values ($2 \leq k \leq 10$) and retain the optimal clustering[2] according to the **silhouette coefficient** (Rousseeuw, 1987). This metric has been previously used for sense induction (Cocos and Callison-Burch, 2016). For a data point p, the silhouette coefficient (SIL) measures the intra-cluster distance $w(p)$ (i.e. the average distance from p to every other data point in the same cluster), and compares it with the inter-cluster distance ($b(p)$), i.e. the average distance of p to all points in its nearest cluster. Equation 5.1 contains the formulas for $w(p)$ and $b(p)$, where d corresponds to the Euclidean distance between p and another data point q. c_p denotes the cluster containing p. Equation 5.2 gives the Silhouette coefficient of a data point p.

$$w(i) = \frac{\sum_{q \epsilon c_p p \neq q} d(p,q)}{|c_p| - 1} \qquad b(p) = \min_{c_q \neq c_p} \frac{\sum_{q \epsilon c_q} d(p,q)}{|c_q|} \tag{5.1}$$

$$sil(p) = \frac{b(p) - w(p)}{\max(w(p), b(p))} \tag{5.2}$$

The SIL value for a clustering C ranges from -1 to 1 and is obtained by averaging the SIL values calculated for all data points $p \in P$ (Equation 5.3). We retain the k of the clustering with the highest mean SIL.

$$\text{SIL}(C) = \frac{\sum_{p \epsilon P} sil(p)}{|P|} \tag{5.3}$$

Since BERT representations' cosine similarity correlates well with usage similarity (as seen in Chapter 4), we also use pairwise cosine distances obtained from BERT representations for clustering. We perform clustering directly on the cosine distance matrix for a lemma. Since the k-means algorithm needs data points with their coordinates to calculate centroids, we cannot use it on this type of data. Instead, we use **agglomerative clustering** with average linkage (BERT-AGG). For comparison, we also use agglomerative clustering on the gold usage similarity scores from the Usim dataset, transformed into distances (`Gold-AGG`).

5.2.2.2 Clusterability metrics

We predict the clusterability of a target word by measuring the quality of its clustering, using the Separability (SEP) and Variance Ratio (VR) metrics (Ackerman and Ben-David, 2009), the two best-performing metrics in McCarthy et al. (2016). We also apply two more measures for clusterability estimation: the silhouette coefficient (which provides estimates of clustering quality) and Dip's test.

[2]The `scikit-learn` implementation of k-means that we use in our experiments runs 10 iterations of each clustering with different seeds by default, and returns the best clustering according to the loss (the sum of squared distances of data points to their closest cluster center).

- **Variance Ratio** (VR) (Zhang, 2001). VR calculates the ratio of the within- and between-cluster variance for a given clustering solution. First, the variance of a cluster y is calculated:

$$\sigma^2(Y) = \frac{1}{|y|}\sum_{p\epsilon y}(y_p - \bar{y})^2 \tag{5.4}$$

where $\bar{y}$ denotes the centroid of cluster y. Then the within-cluster variance W and the between-cluster variance B of a clustering solution C are calculated in the following way:

$$W(C) = \sum_{j=1}^{k} p_j \sigma^2(x_j) \tag{5.5}$$

$$B(C) = \sum_{j=1}^{k} r_j(\bar{x}_j - \bar{x})^2 \tag{5.6}$$

where k is the number of clusters, x is the set of all data points and $r_j = \frac{|x_j|}{|x|}$. x_j are the data points in cluster j. Finally, the VR of a clustering C is obtained as the ratio between $B(C)$ and $W(C)$:

$$VR = \frac{B(C)}{W(C)} \tag{5.7}$$

- **Separability** (SEP) (Ostrovsky et al., 2012). SEP measures the difference in loss between clustering with $k-1$ and k clusters. We use k-means' sum of squared distances (SS) of data points to their closest cluster center as the loss. In an optimal clustering C_k of the dataset x with k clusters, SEP is defined as follows:

$$SEP(x,k) = \frac{loss(C_k)}{loss(C_{k-1})} \tag{5.8}$$

- **Dip's test** (DIP). Dip's test is a statistical test which is used to determine if a distribution is multimodal, i.e. whether it has multiple peaks or modes. In a highly clusterable dataset, pairwise distances are very short for similar datapoints and very long if they belong to different groups (Adolfsson et al., 2019). Therefore, their distribution is expected to be at least bimodal. On the contrary, in less clusterable data distances are more evenly distributed. DIP determines whether a distribution is multimodal or not by comparing it to a unimodal distribution (Hartigan et al., 1985). We use the p-value given by this test, which indicates the probability of observing the given distance distribution based on the null hypothesis that it comes from a unimodal distribution. The smaller the p-value, the more multimodal (and clusterable) the dataset is. This measure differs from the previous ones in that no preliminary clustering step is required.

For VR and SIL, a higher value indicates higher clusterability. The opposite holds for SEP and DIP, where a higher value indicates lower clusterability. VR and SEP require calculating cluster centroids. When we perform agglomerative clustering (BERT-AGG), which does not rely on the BERT vectors themselves but on the cosine distance matrix, we use the corresponding BERT representations to calculate the cluster centroids for these two metrics.

5.3 Evaluation

We measure the clusterability of words in the same dataset that was used in the work of McCarthy et al. (2016). This is the Usim dataset (Erk et al., 2009, 2013), which contains pairwise manual usage similarity annotations for 56 words. McCarthy et al. (2016) derive two gold standard clusterability metrics from Usim:

- **Uiaa** is the inter-annotator agreement for a lemma in terms of average pairwise Spearman's correlation between annotators' judgments. Higher Uiaa values indicate higher clusterability, meaning that sense partitions are clearer and easier to agree upon.
- **Umid** is the proportion of mid-range judgments (between 2 and 4) assigned by annotators to all sentences of a target word. It indicates how often usages do not have identical (5) or completely different (1) meaning. Therefore, higher values indicate lower clusterability.

We calculate Spearman's ρ correlation between the predictions of each clusterability metric and the Uiaa and Umid measures. We also compare to results obtained using McCarthy et al. (2016)'s manual substitute-based representations. Their study included only 45 lemmas in Usim for which both substitute (McCarthy and Navigli, 2007) and translation annotations (Mihalcea et al., 2010) were available. To ease comparison, we re-implemented their model with manual substitutes (Manual-SUB with the graph-partitioning k-selection method) and applied it to all 56 words in Usim, as substitutes are available for all target words in the dataset. We also report results obtained by Manual-SUB representations using our k-selection and clusterability metrics.

5.4 Results

Table 5.1 contains the correlation scores obtained between clusterability values and the gold partitionability estimates. The top part of the table shows results using contextualised representations (-REP) and and distance matrices (-AGG). The best layers for BERT and ELMo are indicated as subscripts. In the lower part of the table we provide results with substitute-based representations.

Agglomerative clustering on the gold Usim similarity scores (Gold-AGG) gives the best results on the Uiaa evaluation in combination with the SIL clusterability metric ($\rho = 0.80$). This is unsurprising, since Umid and Uiaa are derived from the same Usim scores. From our automatically generated representations, the strongest correlation with Uiaa (0.69) is

Gold	Metric	BERT-REP	c2v-REP	ELMo-REP	BERT-AGG	Gold-AGG
Uiaa	SIL ↗	0.61*$_{11}$	0.06	0.21$_{2}$	**0.69***$_{10}$	**0.80***
	VR ↗	0.17$_{12}$	0.14	0.19$_{2}$	0.33*$_{12}$	–
	SEP ↘	-0.48*$_{10}$	-0.12	-0.24$_{2}$	-0.48*$_{11}$	–
	DIP ↘	-0.14$_{12}$	0.13	-0.01$_{3}$	-0.21$_{11}$	-0.14
Umid	SIL ↘	**-0.46***$_{10}$	0.05	-0.06$_{2}$	-0.44*$_{8}$	**-0.48***
	VR ↘	-0.24$_{9}$	-0.08	-0.15$_{3}$	-0.32*$_{5}$	–
	SEP ↗	0.43*$_{9}$	-0.01	0.08$_{3}$	0.43*$_{9}$	–
	DIP ↗	0.26$_{7}$	-0.18	0.16$_{3}$	0.19$_{11}$	0.27
Gold	**Metric**	**c2v-SUB**	**BERT-SUB**	**Manual-SUB**	**c2v-SUBVECS**	**BERT-SUBVECS**
Uiaa	SIL ↗	-0.06	0.12	0.32*	-0.09	0.43*
	VR ↗	-0.10	0.14	0.34*	-0.12	0.27*
	SEP ↘	0.06	-0.11	-0.20	0.08	-0.47*
	DIP ↘	0.15	0.17	-0.29	-0.01	0.11
Umid	SIL ↘	-0.21	-0.07	-0.38*	-0.10	-0.36*
	VR ↘	-0.12	-0.04	-0.24	-0.01	-0.19
	SEP ↗	0.16	0.06	0.16	0.14	0.43*
	DIP ↗	-0.09	-0.03	0.32	0.03	-0.12

Table 5.1: Spearman's ρ correlation between automatic clusterability metrics and the gold standard partitionability estimates, Uiaa and Umid. Significant correlations (where the null hypothesis $\rho = 0$ is rejected with $\alpha < 0.05$) are marked with *. The arrows indicate the expected direction of correlation for each metric. Subscripts for BERT and ELMo indicate the layer of the representations that achieved best performance. The top part of the table contains results with contextualised representations and cosine distances, and the lower part shows results of substitute-based representations.

obtained with BERT-AGG and the SIL clusterability metric. The SIL metric also works well with BERT-REP achieving the strongest correlation with Umid (-0.46). SIL constitutes, thus, a good alternative to the SEP and VR metrics used in previous studies when combined with BERT-based representations.

Interestingly, the correlations obtained using raw BERT contextualised representations are much higher than the ones observed with representations relying on manual substitutes (Manual-SUB). These were in the range of 0.20-0.34 for Uiaa and 0.16-0.38 for Umid (in absolute value). Table 5.2 contains the results obtained with the re-implementation of McCarthy et al. (2016)'s method (using graph-partitioning to select k) on the 56 target words in Usim. These results show that BERT representations offer good estimates of the partitionability of words into senses, improving over manual substitute annotations. On the other hand, ELMo and especially context2vec representations obtain much

Metric	Umid	Uiaa
SEP	0.31*	-0.27*
VR	-0.22	0.27*

Table 5.2: Spearman's ρ correlations between gold standard estimates for the 56 Usim words and clusterability metrics, using Manual-SUB representations and the (McCarthy et al., 2016)'s graph-partitioning method to select the number of clusters.

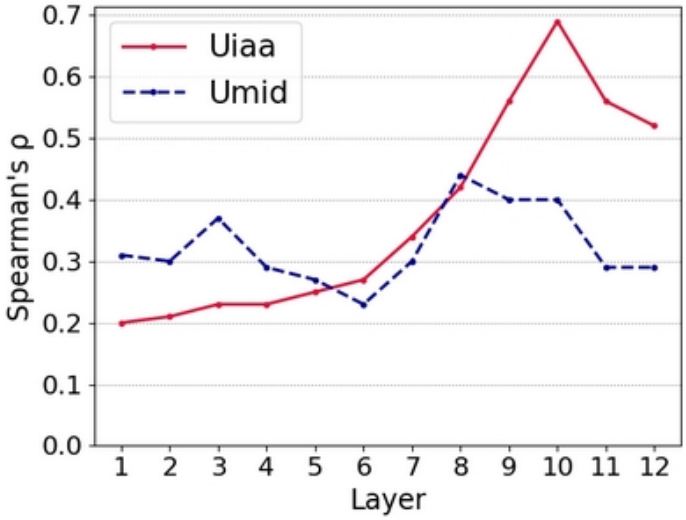


Figure 5.2: Spearman's ρ correlations between the gold standard Umid and Uiaa measures, and clusterability estimates obtained using agglomerative clustering on a cosine distance matrix of BERT representations at different layers.

poorer results on this task. The strongest correlations they achieve are -0.24 and 0.14.

As expected, the substitution-based approach performs better with clean manual substitutes (Manual-SUB) than with automatically generated ones (BERT-SUB, c2v-SUB). Representations based on automatic substitutes do not perform well, even when using BERT-based substitution. This is probably due to the lower quality and bigger size of the PPDB substitute pool. Despite this, taking the average of BERT (avg (4)) representations for substitutes proposed by BERT at each instance (BERT-SUBVECS) proves to be useful for word clusterability estimation, and leads to better results than Manual-SUB. SUB vectors are sparse; they rely solely on substitute overlap and contain no distributional semantic information. This result shows the benefit of including distributional knowledge, which compensates the poorer quality of automatic substitute annotations compared to manual annotations.

Among all clusterability metrics, SIL gives best results overall. The other proposed metric, Dip's test (DIP), obtains the worst overall results, which are sometimes in the opposite direction than expected (for example, with c2v-REP).

We present a per layer analysis of the correlations obtained with the best performing BERT representations (BERT-AGG) and the SIL metric in Figure 5.2. We report the absolute values of the correlation coefficient for a more straightforward comparison. For Uiaa, the higher layers of the model make the best predictions. Similarly to what we observed in our usage similarity prediction experiments (cf. Section 4.5), correlations increase monotonically up to layer 10, and then they slightly decrease. Umid prediction shows a more irregular pattern: it peaks at layers 3 and 8, and decreases again in the last layers. We also report the individual clusterability values obtained for each lemma with the best method (BERT-AGG), along with their Uiaa and Umid scores, in Appendix A.2.1.

Figure 5.3 shows a PCA visualisation of BERT representations for two non-clusterable words (*work.v* and *new.a*) and two highly clusterable words (*charge.v* and *fire.v*), according to

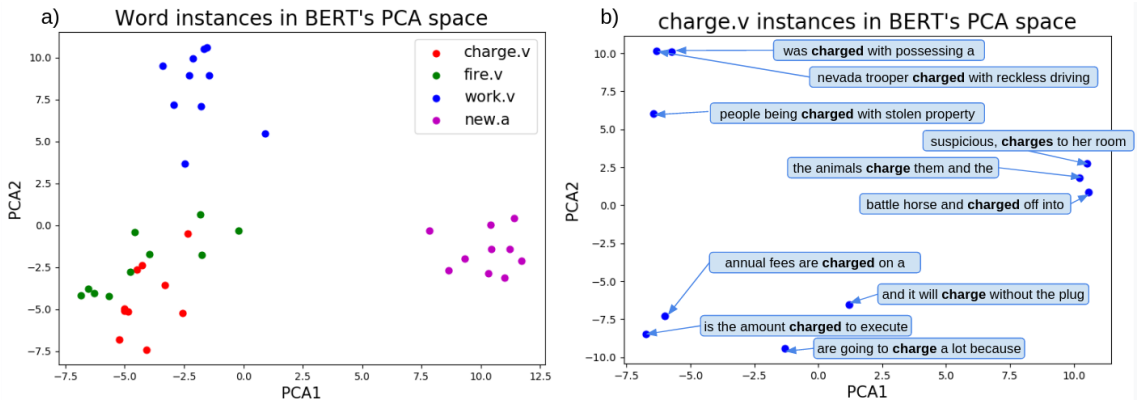


Figure 5.3: PCA visualisation of BERT representations from the 10th layer of Usim instances of (a) *charge.v*, *fire.v*, *work.v* and *new.a*; and (b) instances of the clusterable word *charge.v*, with their sentential context.

Uiaa. We observe that *new.a* presents no clearly clusterable structure, whereas *charge.v* and *fire.v* have some distinguishable clusters. *Work.v* has a low clusterability value, but higher than that of *new.a*.

5.5 Modifying Representations of Clusterable Words

We want to explore whether we can distinguish lemmas for which different types of representations would be preferable, e.g. at the token- or sense-level. We hypothesise that clusterable words, with clear sense boundaries, do not need to be assigned fine-grained instance-level representations and may benefit from a higher level of abstraction. Concretely, our goal is to investigate whether using a multi-prototype approach for clusterable words (keeping instance representations of non-clusterable words unchanged) would result in better semantic representations. Importantly, we want to see if automatic clusterability estimations, obtained from unrestricted text, can be used to determine what words should undergo this modification.

We propose a way to modify representations of clusterable words according to automatic clusterability estimations and evaluate the modified representations on the WiC dataset, where a model has to determine, for two instances of a word, whether they are used in the same sense. We expect the notion of clusterability to be relevant for this task; and if multi-prototype representations are more adequate for clusterable words, we expect to see an improvement on WiC. We begin with an experiment to verify the importance of clusterability for solving the WiC task. Specifically, we compare the performance of BERT on WiC instances involving clusterable and non-clusterable words, according to gold clusterability judgments (Section 5.5.1). Then, we describe how we scale up clusterability estimation, clustering BERT representations of instances of new words in a bigger corpus. We also present our observations on the clusters obtained (Section 5.5.2). Finally, we propose a simple way of turning token-level representations of clusterable words (according to automatic predictions) into multi-prototype representations, and evaluate this approach on WiC (Section 5.5.3).

5.5.1 Impact of Words' Clusterability on Usage Similarity Predictions

We carry out an initial analysis of the performance of BERT on the WiC 1.0 dataset comparing the results on clusterable vs non-clusterable words according to gold clusterability estimates (Uiaa and Umid) from Usim. This analysis allows us to assess the impact that the clusterability level of words has on BERT's performance on this task. We expect performance to be lower on instance pairs involving less clusterable words because it is harder to determine whether they belong to the same sense. We analyse the results obtained by BERT for different words in WiC in the light of their gold clusterability values.

Note that Usim only contains 10 sentences per target word, which may not always constitute a representative sample of its possible contexts. However, sentences were selected manually for 26 out of the 56 words in Usim to ensure a variety of senses, and the usage similarity scores (on which Uiaa and Umid are based) come from manual annotations. Therefore, we consider the gold partitionability judgments to be good enough for the analysis described in this section.

We train a logistic regression classifier on all training instances in WiC 1.0 that do not involve Usim target words (5,125 sentence pairs), using cosine similarity from BERT representations at the 10th layer as the only feature. We evaluate the model on instances from the training and development sets that involve one of the target words in Usim (308 pairs). We define a threshold T of clusterability values which serves to separate words into clusterable and non-clusterable. For example, for Uiaa, a word is considered to be clusterable if its Uiaa value is equal or above a threshold T, and words with a Uiaa score $< T$ are considered to be non-clusterable. We compare BERT's performance on clusterable vs non-clusterable words across different thresholds T based on Uiaa and Umid.

Results of this experiment are shown in Figure 5.4. We see that BERT systematically performs better on clusterable words (according to these gold clusterability estimates) than on non-clusterable words. This is not surprising, as clusterable words have, by definition, clearer boundaries between senses and it is therefore easier to decide whether two instances belong to the same sense or not. This result is in line with what we observed in the previous chapter (Section 4.5), where BERT performance on the Usim dataset correlated with the Uiaa and Umid measures. This result confirms that clusterability has an impact on model performance on usage similarity estimation, and thus justifies using WiC for evaluating our modified representations (Section 5.5.3). It, however, also highlights the fact that there's more room for improvement in representations of words with fine-grained distinctions than of clearly ambiguous words.[3]

[3] An important reason why words with fine-grained distinctions present a bigger challenge is that there are multiple valid ways of partitioning them into senses, and the partitions present in WiC (based on WordNet (Miller et al., 1993) and other resources) may or may not be relevant in another task (Kilgarriff, 1997).

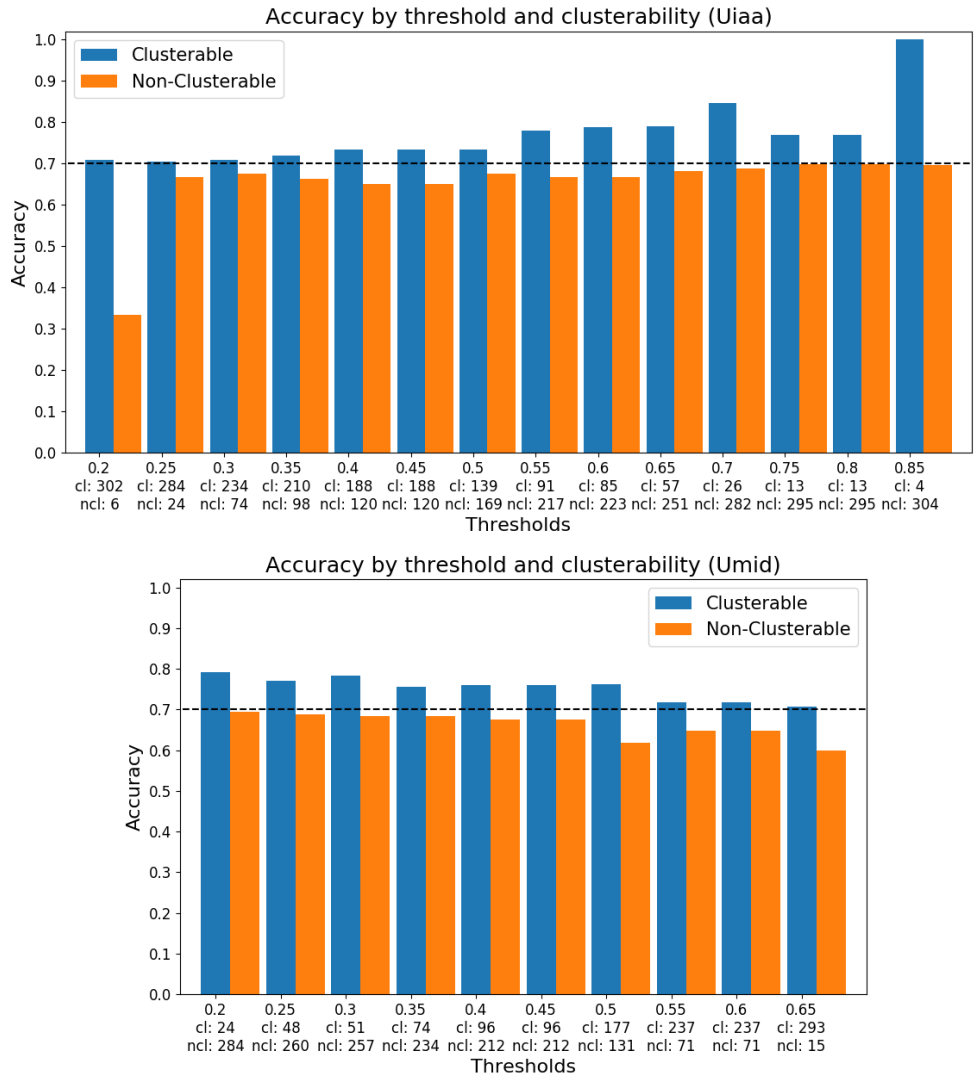


Figure 5.4: Accuracy obtained with BERT representations on WiC instances involving Usim words. We show results separately for clusterable and non-clusterable words across clusterability thresholds (x axis). The clusterability values used are Uiaa (top) and Umid (bottom). *cl* and *ncl* refer to the number of clusterable and non-clusterable words with each threshold.

5.5.2 Scaling up Clusterability Estimation

BERT representations have given good results on clusterability estimation (Section 5.4) on the Usim dataset. The approach is not restricted to manual annotations and can therefore be used to obtain predictions from unrestricted text and for more words in the vocabulary. In order to modify representations of clusterable words and evaluate them on the WiC dataset, we want to obtain clusterability estimations for words in WiC. The obtained values will serve, in Section 5.5.3, to determine what words should be represented with a multi-prototype approach. In this section, we describe how we obtain clusterability estimations for WiC words from a bigger corpus. Specifically, we cluster instances of WiC words in a corpus and calculate their clusterability values. We also present a qualitative analysis of the clusters proposed by BERT

on this data.

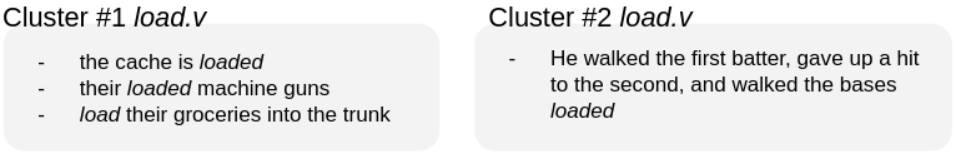
We choose the 20 Newsgroups dataset[4] for its variety of topics and its moderate size. This corpus contains 18,846 newsgroups posts on 20 different subjects, including sports, politics, electronics, and others, with around 6M words in total. We pre-process the corpus removing headers, footers and quotation blocks. We split it into sentences and perform lemmatisation and pos-tagging.[5] We extract sentences from this corpus for 1,519 target words in WiC 1.0. We only consider words for which at least 10 sentences are available in 20 Newsgroups, and use at most 1,000 sentences per target word for clustering. The average of sentences available per word is 160.

We use the best clusterability method for obtaining a clusterability estimate for every word from these sentences: we apply agglomerative clustering to the cosine distance matrix obtained from representations in the 10th layer of BERT, and use the silhouette score as a clusterability metric. We observe that the clusterability values estimated from these sentences are overall higher than those obtained in the experiment on Usim. Those ranged between 0.12 (for the least clusterable word, *new.a*) and 0.44 (for the most clusterable word, *fire.v*), with a mean of 0.23. The new values range from 0.11 (for *describe.v*) to 0.70 (for *void.n*), with a mean of 0.32.

An exploration of the sentences in the proposed clusters, and of the new clusterability estimations, reveals some interesting properties and behaviour of BERT representations. We provide examples that illustrate the meaning expressed in several clusters.

BERT embeddings seem to be quite sensitive to collocational phenomena. Consider the following examples with the words *speak.v* and *load.v*. In the case of *speak.v*, the expression "so to speak" and similar expressions containing the word "speaking" are clustered together:

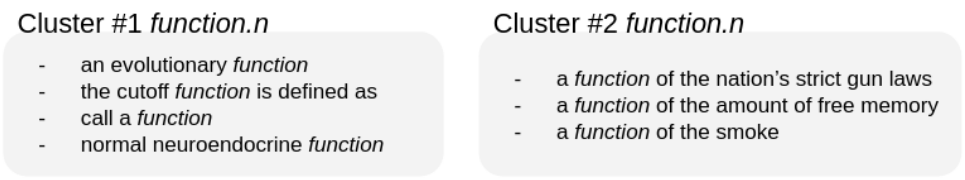
With *load.v*, we find a very distinct sense of the word expressed with a collocation (the baseball term "bases loaded"). This instance forms a cluster on its own, and other senses of *load* all fall into a single cluster:



A similar situation occurs with *function.n*, where sentences that contain the expression "a *function* of" form one cluster, but other semantic distinctions (for example, a programming function and a bodily function) are not captured:

[4] http://qwone.com/~jason/20Newsgroups/, available on `scikit-learn`.

[5] We use the `nltk` and `spacy` libraries.



It is also worth looking at the SIL values obtained. Contradicting our expectations, the word with vague semantics *thing.n* obtains a higher clusterability value (SIL = 0.32) than a word with distinct senses and semantically-motivated clusters like *charge.v* (SIL = 0.28). For *thing.n*, BERT proposes two clusters, one with the expressions "next *thing* you know" and "first *thing* on the morning" and another one with all other instances of *thing*. This, together with the examples shown, could indicate that such collocational phenomena sometimes have a stronger impact on BERT representations than semantic distinctions.

We also note that BERT representations are sometimes clustered according to morphology. In the case of *formula.n*, which is split into 9 clusters, one cluster groups all instances of the plural form *formulae*. This particular case can be explained by the fact that BERT has dedicated wordpieces for *formula* and *formulas*, but not for *formulae*. The approach of averaging all wordpieces of a word probably results in distinct representations in this case, which are assigned their own separate cluster.

The quality of the sentences used also plays a role and may cause some words to have a higher clusterability value than they should. For example, one of the clusters of the word *heart.n* is in fact a misspelling of *heard*; and for the word *die.v*, one of the clusters corresponds to sentences in German containing the German article *die*. It is also important to note that, as mentioned in Section 5.5.1, Usim sentences for 26 out of 56 words were carefully selected to ensure a balance in senses. In 20 Newsgroups, sentences are probably skewed towards the most frequent sense of a word (Kilgarriff, 2004) and they may not contain instances of all senses of a word. This probably contributes to the lower quality of the clusters and clusterability estimations from this corpus.

Not all clusters proposed by BERT present the problems described in this section; we also find cases where word clusterings align very well with our intuitions. However, we believe these cases highlight BERT's sensitivity to certain kinds of contextual information, sometimes to the detriment of semantic information; and they reflect the impact of the quality of the data on the obtained clusters.

5.5.3 Evaluation

Having obtained clusterability estimates for words in WiC (Section 5.5.2), we carry out a simple experiment on the WiC 1.0 dataset to test the benefit that could be derived from modifying word representations based on clusterability information. We consider words with a silhouette coefficient above or equal a certain threshold T to be clusterable. We replace the BERT representations of clusterable words with the centroid of their closest cluster, from those obtained from 20 Newsgroups sentences. This is a typical way of disambiguating a word in

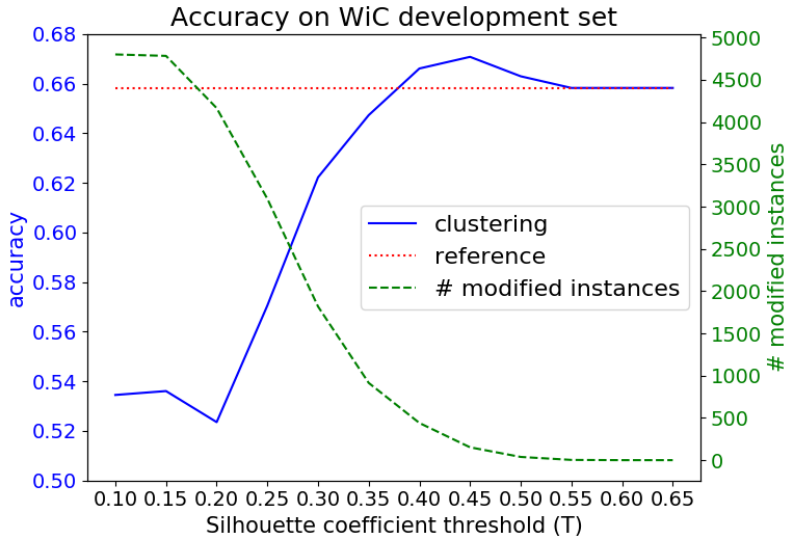


Figure 5.5: Accuracy on the WiC development set when using cluster centroids to represent clusterable words (blue line) according to a silhouette coefficient threshold (x axis). The green line shows the number of WiC training sentence pairs that were modified with each threshold. The reference accuracy (red line) corresponds to a model where no representations are modified.

multi-prototype embedding approaches (Huang et al., 2012). The closest cluster is determined based on the cosine similarity of the word instance representation to each cluster centroid. Representations of words that are not considered to be clusterable (with a silhouette score $< T$) are not modified. We use representations from the 10th layer of the BERT model, as we did for clustering. We train a logistic regression classifier on the WiC training set using as single feature the cosine similarity between the two word instance representations in each sentence pair. We test different values for the threshold T and compare the model's performance to that of a reference model where the representations are not modified. We evaluate the models on the development set.

Results for this experiment are found in Figure 5.5. We observe a slight improvement over the reference accuracy (0.658) when using a threshold T of 0.40, 0.45 or 0.50. The highest accuracy obtained is 0.671 with $T = 0.45$ when modifying 152 out of 6066 WiC instances.[6] For all other clusterability thresholds, the performance is much lower than that of the reference model. This approach requires extracting, storing and clustering BERT representations for a large number of word instances. The performance gain is very limited, considering this high pre-processing cost.

5.6 Discussion and Conclusion

We proposed fully automatic methods for estimating the clusterability of words into senses. We experimented with different types of representations from pre-trained LMs and with substitute-

[6]The WiC training and development sets contain a total of 6066 sentence pairs, with 1791 unique target words (with their part of speech). We have clusters for 937 of these target words, which amount to 5212 instances in the two subsets.

based representations based on automatic substitutes, and also proposed two new clusterability metrics. We found that the best method, based on BERT cosine similarities, correlates better with human clusterability estimates than previous approaches based on manual annotations.

Using the best-performing approach, we clustered word instances and obtained automatic clusterability estimates from a larger corpus. We used these predictions to inform a method that modifies BERT representations of clusterable words, turning them into multi-prototype representations.

The qualitative analysis of the clusters and clusterability estimates obtained on the bigger corpus made apparent the difficulty of scaling clusterability estimation to an open vocabulary and free text. The clusters proposed by BERT representations are not always driven by semantic criteria, and are very sensitive to collocational or contextual differences in the usage of words. Our first attempt at modifying representations of clusterable words based on these estimates showed a slight improvement over standard BERT representations, at a high pre-processing cost.

While we see several directions for potential improvement (for instance, obtaining cleaner sentences to improve clusterability estimates, experimenting with other clustering algorithms, or trying methods other than the centroid for modifying representations), we decided to focus on improving BERT's sensitivity to semantic distinctions in general. We believe this is a more promising direction to improving representations than modifying them according to the clusterability level of words, where we observed small gains. Doing this could potentially be beneficial for the representation of more fine-grained distinctions, which are inherently harder to capture, as reflected in our analysis on the WiC dataset using gold clusterability estimations and BERT representations. It could also improve the semantic quality of the clusters obtained with BERT representations.

Inspired by a recent strand of work on injecting different kinds of linguistic information into the BERT model (Arase and Tsujii, 2019; Lauscher et al., 2019), in the next chapter we shift our focus to fine-tuning for making BERT more sensitive to lexical meaning.

Chapter 6

Fine-tuning BERT for Lexical Meaning

6.1 Introduction

In Chapter 4, we have seen that the similarities derived from BERT representations provide quality estimations of word usage similarity. At the same time, however, these representations seem to be highly sensitive to specific contexts of use and to factors other than word meaning, as we observed in the quality of the clusterings of BERT representations in Chapter 5. These observations motivated us to explore this behaviour further, in order to understand the kinds of knowledge BERT is sensitive to; and to focus on improving the model's sensitivity to lexical meaning specifically.

In this chapter, we first analyse the similarities of BERT representations in sentence pairs that differ in specific linguistic phenomena. Recent studies have proposed injecting different kinds of knowledge into deep LMs to make them more sensitive to specific phenomena (Lauscher et al., 2019; Arase and Tsujii, 2019; Shi et al., 2019). This line of work follows from early approaches for improving the semantic quality of static word representations by incorporating knowledge from external lexical resources (Faruqui et al., 2015; Vulić and Mrkšić, 2018). There is also evidence on the superiority of fine-tuning BERT over using its extracted, so-called "frozen" representations for downstream tasks (Peters et al., 2019b). Inspired by this work, we propose to inject lexical semantic knowledge into BERT. We do so by fine-tuning the model on existing semantically annotated datasets and using automatically generated substitutes in context. We fine-tune BERT models for English and Finnish, and evaluate the quality of the resulting representations on the CoSimLex dataset (Armendariz et al., 2020a). This dataset addresses in-context word similarity in multiple languages, and is designed for exploring the effect of context on word meaning in a continuous, or graded, fashion.

Our experiments allow us to learn more about the different kinds of information reflected in BERT representations. We can also gauge the impact of model fine-tuning on the similarity estimates derived from the representations. Importantly, we compare the utility of different fine-tuning tasks, built with manual and automatic semantic annotations.

6.2 Impact of Linguistic Phenomena on BERT Representations

In this section, we explore the impact of different linguistic transformations on the usage similarity estimates that can be drawn from BERT representations. We carry out a comparison of the BERT similarity values obtained between sentences that differ in a specific, controlled linguistic phenomenon. We want to investigate if, and to what extent, transformations that do not change the meaning of a sentence can affect usage similarity values (which would be 1 in two identical sentences). This analysis will provide a clearer picture of the kinds of linguistic phenomena that influence the representations.

We use the SICK dataset (Marelli et al., 2014), a collection of 9,840 English sentence pairs (c_1, c_2) that illustrate different types of transformations. In a sentence pair, c_2 is a transformed version of c_1. This dataset was originally developed to test for compositionality in distributional models, and contains pairwise similarity scores and entailment judgments. However, in our analysis we will only be using the transformation label, which determines whether the meaning of the sentence is preserved. There are three major kinds of transformations in SICK, depending on the effect that they have on sentence meaning: those that create a sentence c_2 with (a) a meaning similar to that of c_1; (b) a meaning that contradicts c_1; and (c) a meaning different from that of c_1, but preserving a high lexical overlap. Table 6.1 contains examples of the nine (out of 12) most represented transformations in the dataset. These include, for example, the transformation of an active sentence to passive voice, the substitution of a word in the sentence by its antonym, or "word scrambling". "Scrambling" involves rearranging words in a sentence, possibly changing their part of speech or the sense used, causing a change in sentence meaning. In our analysis, we use the nine transformations included in the Table.[1]

For each sentence pair (c_1, c_2), containing the sets of words W_{c_1} and W_{c_2}, we collect the BERT representation of the words that are common in c_1 and c_2 ($W_{c_{1-2}} : \{\forall w \in W_{c_1}; w \in W_{c_2}\}$), excluding stop words.[2] For example, in the first sentence in Table 6.1, the common words are *girl*, *strange*, *outfit* and *bike*. For each common word $w \in W_{c_{1-2}}$ (e.g. *girl*), we calculate the cosine similarity between its instance in c_1 and in c_2. Finally, we calculate the average of the similarities obtained for each type of transformation. This reflects how much word representations change due to a specific type of transformation. We use the last layer of the `bert-base-uncased` model, as this is the layer on top of which classifiers are placed for fine-tuning.

Results are presented in Figure 6.1. Word scrambling is the transformation that affects representations the most. This is expected, because the meaning of the sentence is not preserved and words may change their form, part of speech and meaning. This is reflected in the low similarity between representations acquired from the original and the transformed sentences. We note that different transformations, even the meaning-preserving ones, have a different impact on average similarities. If BERT representations were only influenced by

[1] We exclude the other three transformations from this analysis because they are much less represented in SICK. These are the transformation from passive to active voice (17 pairs), the expansion of agentive nouns (28), and the conversion of compounds into relative clauses (56).

[2] We use `scikit-learn`'s list of English stop words.

Modification	c_1	c_2
Passive voice (281)	A girl in a strange outfit is riding the bike	The bike is being ridden by a girl in a strange outfit
Lexical substitution (847)	A dog is emerging from a lake	An animal is emerging from a lake
Modifier addition (287)	Two people are sitting on a bench	Two people are sitting on a white bench
Adjective expansion (189)	white dog is standing on a grassy hillside	A white dog is standing on a hill covered by grass
Determiner substitution (268)	Two dogs are playing on a beach	Two dogs are playing on the beach
Negation insertion (419)	The person is going into the water	The person is not going into the water
Opposite determiner (608)	A skateboarder is jumping in the air	No skateboarder is jumping in the air
Antonym (933)	An elderly man is sitting on a bench	A young man is sitting on a bench
Word scrambling (377)	Two dogs are playing in the snow	It is snowing on two playing dogs

Table 6.1: Examples of the most common transformations in the SICK dataset. The numbers in parentheses indicate the amount of sentence pairs available for each transformation. The first section of the Table contains transformations that do not modify the meaning of the sentence (a); the middle section shows those that result in a sentence of an opposite meaning (b). The bottom section shows the word scrambling transformation, where a rearrangement of the words results in a different meaning (c).

semantic factors, we would expect all modifications of type (a) to result in similar similarity values. However, as shown in the Figure, this is not the case. In fact, some transformations of type (a) result in lower similarities than those of type (b) (opposite meaning). For example, after word scrambling, the transformation to passive voice has the lowest similarity values. In this case, the arguments of a verb are shifted, but meaning is preserved. Passivisation affects the similarity estimates more than other transformations involving, for example, word substitution, even when a word is replaced with its antonym (which incurs a change in meaning).[3] This is reflected in the higher similarity scores of the "Lexical substitution" and "Antonym" transformations, which affect representations the least. This can in part be explained by the design of BERT's embedding layer. The input embedding consists of the sum of token, position and segment embeddings. This means that the representations of words that we compare contain information about their position in the sentence, and a change in position, like the one that occurs in passivisation, is reflected in the representations.[4] Adjective expansion, where a

[3]We note that, despite being opposite in meaning, antonyms tend to be distributionally similar to each other because they can occur in the same contexts (Lin et al., 2003). Given BERT's cloze-style pre-training task, it is likely that its representations reflect this similarity.

[4]Mickus et al. (2020) examine the effect of segment embeddings on the representations. These have a key role in BERT's Next Sentence Prediction pre-training task, as they mark the first and second sentence of the input sequence differently. They find that tokens have different representations depending on whether they are in the first or second sentence. They note that this could partly be due to position embeddings, which mark the position

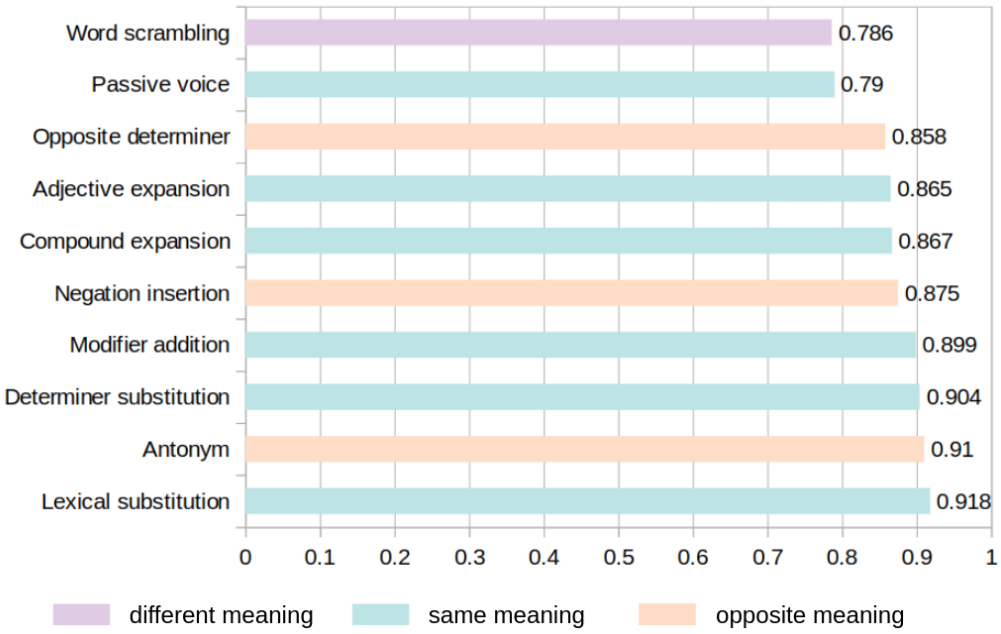


Figure 6.1: Average similarity of BERT representations by transformation type. Representations are extracted from the last layer, and similarities are calculated between instances of the same word. Colours indicate the type of meaning change that each transformation causes.

relative clause is introduced, is also among the transformations that yield the biggest change in representations.

The experiment presented in this section confirms our preliminary observations that BERT word instance representations are strongly influenced by phenomena not strictly related to lexical meaning. We have observed how the usage similarity estimates between two meaning-equivalent sentences decrease when specific kinds of transformations are applied. The effect of these transformations is in some cases bigger than the effect observed with transformations that change the meaning of a sentence. This motivates us to try to enhance the lexical semantic information in BERT representations, making them more sensitive to word meaning. We describe these experiments in the following sections.

6.3 The *Graded Word Similarity in Context* Task

The GWCS SemEval task (Armendariz et al., 2020b) introduced the CoSimLex dataset (Armendariz et al., 2020a), described in more detail in Section 2.1.3.3. The task is focused on the effect of context on human perception of similarity as a graded notion, in contrast to the WiC dataset. CoSimLex differs from the Usim dataset in three respects: it presents word pairs within the same context; it addresses the similarity of instances of different words; and it is available in multiple languages: English, Croatian, Slovene and Finnish. GWSC consisted of two subtasks where models had to predict (1) the shift in meaning similarity for a pair of words (w_a, w_b)

of a token in the entire input sequence, not in an individual segment. In our analysis, however, we use only the first segment, as we input individual sentences.

from one context to another, and (2) the similarity of two word instances in the same context. This is illustrated by sentences c_1 and c_2, two contexts where *body* and *chest* co-occur.

c_1 (...) The International Labour Office (ILO) is the Organization's research body and publishing house. Since 1950, the ILO has periodically published guidelines on how to classify chest X-rays for pneumoconiosis.

c_2 The dance is performed by moving one's shoulders up and down with arms bent toward the chest. Then one rocks the upper body back and forth (...)

A change in meaning similarity occurs between the highlighted words in the two sentences. *Chest* denotes a part of the human body in the two cases. The words are less similar in context c_1, where *body* refers to an organisation, than in context c_2 where both words refer to the human anatomy. The shift in meaning is reflected in the difference between gold similarity scores assigned to these instance pairs in the GWSC dataset (1.83 vs. 6.51). In subtask 1, the difference in values has to be predicted ($6.51 - 1.83 = 4.68$). In subtask 2, models must predict the similarity scores themselves (1.83 and 6.51). All predictions are evaluated against gold judgments provided by annotators. There are 340 context pairs available for English, 112 for Croatian, 11 for Slovene and 24 for Finnish; which were used for evaluation. In addition to that, 10 sentence pairs were released as trial data for all languages but Finnish. We participated in the English and Finnish tasks with our fine-tuned models described in the following section as a way of evaluating their lexical semantic quality.

6.4 System Overview

6.4.1 Background

Our methodology draws inspiration from recent work on injecting semantic information into pre-trained language models (Lauscher et al., 2019; Arase and Tsujii, 2019; Shi et al., 2019; Peters et al., 2019a; Qu et al., 2019; Levine et al., 2020). This can be done at two stages: during model pre-training or during fine-tuning.

Lauscher et al. (2019) opt for the first, adding a lexical task to BERT's two training objectives (language modelling and next sentence prediction). They pre-train a smaller BERT model from scratch with a binary word relation classification task. Specifically, they feed the model with word pairs and the model has to learn whether they stand in some lexical relation, such as synonymy or hyponymy. The semantic knowledge used in this additional task comes from pre-defined lexicographic resources (like WordNet (Miller, 1995)). This is shown to be beneficial in almost all tasks in the GLUE benchmark (Wang et al., 2018)[5] compared to a BERT model of the same size trained without this task.

[5]The GLUE (General Language Understanding Evaluation) benchmark is a set of nine tasks, with their corresponding datasets, targeting different aspects of Natural Language Understanding. The tasks involve Natural Language Inference, Sentiment Analysis and Semantic Textual Similarity, among others.

Arase and Tsujii (2019) inject semantic knowledge into BERT by fine-tuning the pre-trained model on paraphrase data. Their method consists in simultaneously learning to discriminate phrasal and sentential paraphrases, using two separate classification heads. They subsequently fine-tune the model for a second time for the related tasks of paraphrase identification and semantic equivalence assessment, and report results that demonstrate improved performance over a model that has not been exposed to paraphrase data. We follow their approach, which they refer to as "transfer fine-tuning". We fine-tune BERT models for English and Finnish on a set of semantic tasks that are closely related to the GWSC task, since no training data is available for GWSC. Our goal is improve the semantic knowledge in BERT representations by first exposing the model to another lexical semantic task.

One of the tasks we use for fine-tuning is inspired by the retrofitting approach of Shi et al. (2019). They observe that distances between ELMo (Peters et al., 2018a) representations are not always intuitive: the embeddings of two instances of the same word occurring in meaning-equivalent sentences (e.g. *flat* in "Some people believe earth is flat. Why?" and "Why do people still believe in flat earth?") are sometimes farther apart than representations of antonyms (*large* and *small*) in sentences with different meanings. They propose an orthogonal transformation for ELMo that is trained to bring representations of word instances closer when they appear in sentences that have the same meaning. They collect sentence pairs from the Microsoft Research Paraphrase Corpus (MRPC) (Dolan et al., 2004) that share a word and which are paraphrases of each other (T) or not (F). They show that this retrofitting approach improves ELMo's performance in a wide range of semantic tasks at the sentence level (sentiment analysis, inference and sentence relatedness). We follow their data collection method to obtain word instances for fine-tuning BERT in one of our fine-tuning tasks. We replace MRPC with the Opusparcus resource (Creutz, 2018) since it covers two of the languages addressed in GWSC, English and Finnish.

6.4.2 Datasets

We fine-tune pre-trained BERT models on semantic tasks that are related to GWSC. We select tasks that address the similarity of word meaning in context, and use the corresponding datasets to specialise BERT on this specific aspect of meaning. The Usim, CoInCo and WiC datasets are described in more detail in Section 2.1.3.2 and 2.1.3.3. Table 6.2 contains annotated instances from each dataset used in our experiments.

Usim and CoInCo The Usim dataset contains 10 sentences for each of 56 words of different parts of speech, manually annotated with pairwise usage similarity scores (Erk et al., 2009, 2013). As in GWSC, similarity scores in Usim are graded. To binarise the usage similarity scores and use them for fine-tuning, we consider only sentence pairs annotated with low similarity scores (score < 2) as instances denoting a different meaning (F), and highly similar sentence pairs (score > 4) as instances of the same sense (T). In total, we use 1,399 Usim sentence pairs for fine-tuning. Since this is a small dataset, we combine it with instances from CoInCo (Kremer et al., 2014). We use the CoInCo sentence pairs that we extracted for Usim prediction (Section

4.2), where instance pairs were considered to have the same (T) or a different (F) meaning depending on their substitute overlap. We collect additional data from CoInCo relaxing the class inclusion constraints. Before, we only allowed instances with at least four substitutes. Now, we retain all instances regardless of the number of available substitutes. In Section 4.2, we considered as (T) instance pairs that have at least 75% of substitutes in common, and as F examples pairs that do not share any substitute. Now, we accept as (T) pairs with at least 50% of common substitutes, and as (F) examples pairs that share at most one substitute. We retain up to 500 instance pairs per lemma in CoInCo, when available. We balance the two classes (T and F) and merge the obtained instances with the 5,023 pairs collected in the first place (Section 4.2), removing the duplicates. In total, we have 22,226 CoInCo instance pairs for fine-tuning. We use these instances in combination with the Usim data.

WiC The WiC dataset contains pairs of word instances in context with the same or a different meaning (Pilehvar and Camacho-Collados, 2019). The dataset comes with a train/dev/test split. We use the training set (5,428 sentence pairs) with its labels (T or F) as data for fine-tuning.

ukWaC-subs The GWSC task involves pairs of *different* words that can have similar meanings in some contexts and not in others (e.g. *body* and *chest*). Given that no training data is available, we automatically create one more dataset for fine-tuning called ukWaC-subs, which approximates this task.

ukWaC-subs contains pairs of sentences (c_1, c_2) that differ in one word only. We create the data by substituting a word w in c_1 by either (a) a correct substitute; (b) a word that is a good synonym of w and could have been a correct substitute in another context but not in this one; or (c) a random word of the same part of speech as w. This is illustrated by the three ukWaC-subs sentences in Table 6.2. With (a), we expect BERT to learn that *clear* is being used in its UNDERSTANDABLE sense in this context (illustrated by the substitute *ambiguous*). In (b), we want BERT to learn that despite the (out-of-context) similarity between *present* and *moment*, the latter is not adequate in this context. With (c), we help BERT distinguish *date* from a completely unrelated word (*heritage*). We use this data for a 3-way classification task.

We create this dataset by collecting sentences from the ukWaC corpus (Baroni et al., 2009) and automatically annotating them with lexical substitutes. We identify the content words in a sentence[6] and use as their candidate substitutes their paraphrases in the Paraphrase Database (PPDB) lexical XXL package Ganitkevitch et al. (2013); Pavlick et al. (2015). We only consider as candidates for substitution paraphrase pairs with a PPDB 2.0 score above 2. We then use context2vec embeddings (Melamud et al., 2016) for lexical substitution to rank the candidates according to how well they fit a specific context. We use the *c2vf* method described in the Lexical Substitution Chapter (Section 3.4.2.3), which relies on target-to-substitute and substitute-to-context similarities.

We obtain an ordered ranking R of substitutes $s \in S_t$ for an instance i of a target word t in context C. The highest-ranked substitute is viewed as correct and serves to create instances of

[6]We use only nouns, verbs (excluding modal verbs and auxiliaries), adjectives and adverbs, according to the pos-tags in ukWaC.

Label/ Score	Sentence 1	Sentence 2
	Usim	
T (4.3/5)	We recommend that you **check** with us beforehand.	I have **checked** multiple times with my order and that is not the case.
F (1.3/5)	The romance is uninspiring... and **dry**.	If the mixture is too **dry**, add some water; if it is too soft, add some flour.
	WiC	
T	Laws limit the **sale** of handguns .	They tried to boost **sales**.
F	She didn't want to **answer**.	This may **answer** her needs.
	ukWaC-subs	
a (T)	For neuroscientists, the message was **clear**.	For neuroscientists, the message was **unambiguous**.
b (F)	Need a **present** for someone with a unique name?	Need a **moment** for someone with a unique name?
c (F')	Overdue tasks display on the due **date**.	Overdue tasks display on the due **heritage**.
	Opusparcus	
T	I **love** you so much	I **love** you to the moon and back.
F	yes, Mary, I would **love** to dance.	Why do I **love** him?

Table 6.2: Example instances from each dataset addressing word similarity in context.

type (a). A random word of the same part of speech found in the corpus makes an instance of class (c). To obtain instances of class (b) we could in principle take the last substitute in the ranking. However, due to the noise that exists in PPDB, these often are not correct paraphrases of the target word, even out of context. We therefore apply the PPDB filtering strategy proposed in Section 4.3.2.2 which checks whether substitutes in adjacent positions (s_j, s_{j+1}) in the ranking R form a paraphrase pair in PPDB. If this is not the case for a specific pair, we stop checking at that point in the ranking and retain s_{j+1} as a substitute that represents a different meaning of the target word.

Once the substitutes have been collected, 40% of the instances are assigned to class (a), 30% to class (b) and 30% to (c). One sentence may contain more than one training instance if a substitute ranking is available for different words in it. A training instance is created by replacing the word with the substitute required by the class it has been assigned to, as can be seen in Table 6.2. We create 100,000 instances that we use to fine-tune BERT.

Opusparcus Shi et al. (2019) show that retrofitting ELMo with paraphrases improves its performance on lexical semantic tasks. We follow a similar approach and use paraphrases to fine-tune BERT before applying it to GWSC. We use paraphrases from the Open Subtitles Paraphrase Corpus (Opusparcus) (Creutz, 2018). We use this corpus instead of the Microsoft Research Paraphrase Corpus (Dolan et al., 2004) used by Shi et al. (2019) because it contains paraphrase pairs for six European languages, including English and Finnish which are included in GWSC. Paraphrase pairs in Opusparcus were extracted from movies and TV shows subtitles, and are ranked by quality. We use paraphrases from the Opusparcus training set with a quality score higher than 15,[7] and create our own training instances following the procedure of Shi et al. (2019). Every pair of paraphrases that share a content word constitutes a positive example (T). For every T, we create a negative example (F) by selecting a pair of sentences from the resource which share the same word but are not paraphrases of each other. To avoid creating examples for target words that are highly frequent and have fuzzy semantics, we omit instances of the 200 most frequent words in the Google Books NGram corpus (Michel et al., 2011) (e.g. *make*, *get*, *good*). In total, we use 100,000 sentence pairs from Opusparcus for fine-tuning the English model and 60,520 for Finnish.

6.4.3 Models

We use the five datasets described in the previous section to fine-tune pre-trained BERT models for English and Finnish. All tasks require comparing the meaning of word instances in two different sentences. We form an input sequence (sentence pair) for BERT by joining the two sentences together with the separator token (`[SEP]`) in between. Since the task is at the word level, we do not build our classifier on top of the `[CLS]` token which is an aggregation of the whole input sequence. Instead, our classifier receives as input the BERT representations of the target word instances at the last layer. BERT uses wordpiece tokenisation (Wu et al., 2016), which means that a target word may be split into several tokens. For words that have been split, we average the representations of each wordpiece. We use two kinds of heads for fine-tuning:

- **Classification head**: The representations of the two target tokens are concatenated and fed to a linear classifier which outputs probabilities for each class. We use a cross entropy loss for training. We call this head CLASSIF.
- **Cosine distance head**: We apply the Cosine Embedding Loss (PyTorch, (Paszke et al., 2019)) to the representations of the two target tokens at the last layer. This loss increases the cosine distance of two tokens if they do not have the same meaning, and decreases it in the inverse case. We refer to this head as COSDIST.

Note that the ukWaC-subs dataset is only compatible with the CLASSIF head because it has three classes. To predict the similarity of two target tokens in the GWSC data, we extract their representations from the different layers of a fine-tuned model. We use cosine similarity

[7] Scores range from ∼77 (best quality) to ∼2 (worst quality).

(*cossim*) as our similarity metric. In Subtask 2, which consists in predicting the similarity scores for a pair of words (w_a, w_b) in the same context c, we simply calculate the cosine similarity of their representations in a specific layer ($cossim(w_{a_c}, w_{b_c})$). In Subtask 1, we need to predict a change in similarity between two words w_a and w_b in two different contexts (c_1, c_2). We estimate the change in similarity (ΔSim) with a simple subtraction of the similarities obtained for Subtask 2:

$$\Delta Sim = cossim(w_{a_{c2}}, w_{b_{c2}}) - cossim(w_{a_{c1}}, w_{b_{c1}}) \tag{6.1}$$

where $w_{a_{c2}}$ is the representation of word w_a in context c_2.

6.4.4 Experimental Setup

We participated in GWSC Subtasks 1 and 2 for English and Finnish. We did not address Croatian and Slovenian due to the lack of datasets that could be used for fine-tuning. For English, we fine-tune the `bert-base-uncased` model. For Finnish, we use the uncased Finnish model (`finnish`) (Virtanen et al., 2019)[8] and the uncased Multilingual BERT-base model (`multilingual`).[9] The `finnish` model is trained on 3.3B tokens from different sources including news and Wikipedia text in Finnish. The `multilingual` model was trained on Wikipedia data in 102 languages, but the amount of Finnish training data used is about 30 times smaller than in the `finnish` model (Virtanen et al., 2019). For faster fine-tuning, we set the maximum length to 128 wordpieces and omit examples where a target word occurs after this position.

We use as a development set for English the officially released GWSC trial data (10 sentence pairs) and an earlier release of trial data (8 sentence pairs), both distinct from the test set. We use these data to select the best models and hyperparameters for our official submissions to GWSC. The English test set consists of 340 context pairs for Subtask 1 and 680 unique contexts for Subtask 2. We fine-tune `bert-base-uncased` separately on each of our English datasets for up to 15 epochs. We experiment with the two classification heads {CLASSIF, COSDIST} and with different learning rates {5e-5, 1e-6, 1e-7}. These hyperparameters, along with the layer the word representations are extracted from, are set on the GWSC trial data. Our submitted models were fine-tuned on WiC, Opusparcus and CoInCo-Usim with a learning rate of 5e-5 and 0.1 dropout for 4, 3 and 2 epochs, respectively. The ukWaC-subs model was fine-tuned for 11 epochs with a learning rate of 1e-6 and 0.2 dropout. Dropout was determined based on results on 2,000 held-out ukWaC-subs instances.

Since no trial dataset was released for Finnish, we fixed the hyperparameters for our models to those that worked best for the English Opusparcus data. Our submitted predictions are from the higher layers of the models fine-tuned with the CLASSIF head. The test set for Finnish consists of 24 context pairs in Subtask 1 and 48 unique contexts in Subtask 2.[10]

The metrics used to evaluate model predictions are the uncentered Pearson correlation (ρ) in Subtask 1 (Equation 6.2), and the harmonic mean of Pearson and Spearman correlations ($\bar{\rho}$)

[8] https://github.com/TurkuNLP/FinBERT

[9] https://github.com/google-research/bert/blob/master/multilingual.md

[10] We use HuggingFace's `transformers` library (Wolf et al., 2020) to implement our experiments.

in Subtask 2.

$$CC_{uncentered} = \frac{\sum_{i=1}^{n}(x_i)(y_i)}{\sqrt{(\sum_{i=1}^{n} x_i)^2(\sum_{i=1}^{n} y_i)^2}} \tag{6.2}$$

6.5 Results

Results for the two English and Finnish subtasks are presented in Table 6.3. We report results of the two best systems submitted to each subtask (marked with †) along with results calculated during the post-evaluation phase for comparison. These include baseline predictions made by BERT models without fine-tuning.

Although the two subtasks are highly related, different models perform best in each one. For English, the best result in Subtask 1 (among our official submissions) is obtained by the model fine-tuned on WiC data with the COSDIST head ($\rho = 0.760$). This model occupies the third position in the final ranking and is closely followed by the model fine-tuned on paraphrase data with the CLASSIF head. The best performing model in Subtask 2 is the one fine-tuned on the ukWaC-subs data ($\bar{\rho} = 0.718$) which ranked fourth. The second best model uses the COSDIST head and is trained on the CoInCo and Usim data together. All English models outperform the BERT-based baseline without fine-tuning ($\rho = 0.715$ and $\bar{\rho} = 0.661$). This demonstrates the higher quality of lexical semantic knowledge in our fine-tuned models.

Best results for the Finnish Subtasks 1 and 2 are also produced by different models. The `multilingual` model performs better on Subtask 1 and the `finnish` model on Subtask 2. We observe that similarities assigned to word instance pairs by the `multilingual` model fall in a smaller range (M=0.87, SD=0.04) than those assigned by the `finnish` model (M=0.77, SD=0.07).[11] This explains the low performance of the multilingual model in Subtask 2, where similarity scores have to be predicted. At the same time, however, it does well on Subtask 1 because it captures the magnitude of the difference in similarity between two pairs. Given that no trial data (development set) are available for Finnish and that the maximum number of submissions to the task was nine, we could only try up to five layers per model at submission time. We used the upper layers because they had given better results in English. Our submitted Finnish models, however, perform worse than their counterparts without fine-tuning. The models were ranked sixth and fourth in Subtasks 1 and 2.

During the post-evaluation phase, we had the possibility to test all layers of the models. The sixth layer of the `multilingual` model fine-tuned on Finnish Opusparcus data outperforms the `multilingual` baseline on Subtask 1 ($\rho = 0.718$ vs $\rho = 0.677$), but the other fine-tuned models did not improve over their respective baselines. Surprisingly, the `finnish` baseline model in Subtask 2 ($\bar{\rho} = 0.671$) outperforms the top-ranked model for Finnish among all teams that participated in the task ($\bar{\rho} = 0.645$).

[11] Statistics are taken from layer 11.

Model	Subtask 1	Subtask 2
English		
WiC COSDIST	† 0.760_{11}	0.689_{11}
ukWaC-subs	0.751_{10}	† $\mathbf{0.718}_{10}$
Opusparcus CLASSIF	† 0.751_{11}	0.669_{6}
CoInCo + Usim COSDIST	$\mathbf{0.765}_{10}$	† 0.686_{6}
`bert-base-uncased`	0.715_{11}	0.661_{11}
Finnish		
`multilingual` Opusparcus CLASSIF	† 0.593_{9}	† 0.192_{11}
`multilingual` Opusparcus CLASSIF	$\mathbf{0.718}_{6}$	0.286_{5}
`finnish` Opusparcus CLASSIF	† 0.500_{12}	† 0.491_{9}
`finnish` Opusparcus CLASSIF	0.550_{1}	0.568_{3}
`multilingual`	0.677_{11}	0.388_{9}
`finnish`	0.577_{12}	$\mathbf{0.671}_{12}$

Table 6.3: Results of our English and Finnish models in GWSC Subtasks 1 and 2. The models are compared to three BERT-based baselines without fine-tuning. The evaluation metric in Subtask 1 is Pearson's correlation coefficient. In Subtask 2, it is the harmonic mean of Pearson and Spearman's correlation coefficients. Our official submissions to the GWSC task for each language are marked with †. Subscripts indicate the BERT model layer used.

6.6 Discussion

There are many possible ways in which BERT can be complemented with additional information; in our work, we focus on one of them, fine-tuning. Another approach that we find promising is proposed in Qu et al. (2019). It consists in introducing a word-level feature at the embedding layer, which is added to the token, position and segment embeddings used in BERT and is optimised during fine-tuning on a task that could benefit from this information. In their case, they create a binary feature indicating whether a word has been previously used in a conversation, and find it useful on a conversational Question Answering task. One could potentially include, at that level, more information about the meaning of a word, or other information like its frequency, number of senses, or partitionability, if reliable estimates are available.

Another possibility is to also fine-tune models on tasks related to lexical meaning, but controlling for specific positional, syntactic and collocational phenomena. For example, the dataset for fine-tuning could be built in a way that reduces BERT's sensitivity to these phenomena, including sentences where a word occurs in different morphological forms, grammatical functions, or in different positions in the sentence. Additionally, in light of Mickus et al. (2020)'s finding on the important influence of BERT's sequence segment on word representations, it would also be interesting to fine-tune BERT on tasks where only one segment is used.

Finally, interpretability work (introduced in Section 2.3) can provide insights as to how im-

prove a model. This line of work aims at understanding the inner workings of deep pre-trained LMs, and investigates the linguistic and world knowledge encoded in different layers (Tenney et al., 2019a) or attention heads (Voita et al., 2019b) of the models. We believe understanding how and where BERT makes use of different kinds of information could guide approaches aiming to improving the model. For example, if we identify the layers or attention heads where lexical information is more or less prominent, we could adapt the weight given to representations from different layers accordingly, or prune heads that contain unnecessary information (Michel et al., 2019).

6.7 Conclusion

In this chapter, we explored the impact of different linguistic transformations on BERT representations. Our investigation relied on an exploration of similarity estimates obtained from meaning-equivalent sentences which illustrate controlled linguistic transformations.

We followed recent advances in injecting knowledge into BERT to improve the modelling of lexical semantic knowledge in the representations derived from the model. We investigated the effect of fine-tuning pre-trained BERT models on existing datasets that address word meaning similarity in context. We proposed a novel fine-tuning task where in-context lexical similarity is approximated through automatic substitute annotations. We evaluated this fine-tuning approach in the frame of SemEval 2020 task 3, "Graded Word Similarity in Context" (GWSC), where we participated with models for English and Finnish. Our results with English models demonstrate the benefit of fine-tuning BERT on a task that is closely related to the end task. Results on our model trained on data with automatic substitutions show that this is the case even when data are automatically obtained, and hence of lower quality than hand-crafted data. The lower results of models for Finnish highlight the importance of data availability for fine-tuning, as we could only fine-tune models with paraphrases. We also found that similarity estimates from the multilingual BERT model, at least for the Finnish language, are very high and fall in a narrow range of scores, affecting its results on Subtask 2 in GWSC. Finally, we discussed several relevant alternative ways of injecting knowledge into BERT, including possible modifications of our approach. We also emphasise the utility of insights and methodology coming from interpretability work for improving deep pre-trained LMs.

Chapter 7

Polysemy Level Prediction

7.1 Introduction

In previous chapters, we explored the ability of contextualised representations to capture word meaning in context and proposed ways to make them more sensitive to semantic information. We have also used these representations to predict words' clusterability, a property that reflects the organisation of their semantic space. In this chapter, we focus on another lexical property, the degree of polysemy of words, i.e. their potential to express different meanings.

Words can have one or multiple senses, i.e. they can be monosemous or polysemous. Polysemous words can be situated at a higher or lower polysemy level and express a different number of senses. Apart from its theoretical interest, knowing the polysemy level of words has numerous practical implications: it can point to monosemous words which can be safe cues for disambiguation in running text (Leacock et al., 1998; Agirre and Martinez, 2004; Loureiro and Camacho-Collados, 2020) and determine the needs in terms of context size needed for disambiguation (e.g. in queries, chatbots). Similarly to clusterability (McCarthy et al., 2016) (cf. Chapter 5), it can also be useful for lexicographers to determine the number of entries and senses for a word, and to estimate the effort needed for semantic annotation. Furthermore, it could be used to identify less polysemous words that could guide cross-lingual transfer. Finally, detecting variations in the polysemy level of a word across time is highly relevant for the study of lexical semantic change (Rosenfeld and Erk, 2018; Giulianelli et al., 2020; Schlechtweg et al., 2020).

We want to investigate whether the semantic space of the contextualised representations generated by pre-trained language models reflects this property of words. We also want to discover whether the models' knowledge about polysemy is acquired through exposure to the context of new word instances or during pre-training. In this chapter, we propose methodology to answer these questions about BERT and other pre-trained LMs.

Our approach involves the use of datasets carefully designed to reflect different sense distributions. It also accounts for the strong correlation between word frequency and number of senses (Zipf, 1945), and for the relation of grammatical category and polysemy. Importantly, our investigation encompasses monolingual models in different languages (English, French,

Spanish and Greek) and multilingual BERT.

As discussed in Section 2.3.2, several works investigate the knowledge that pre-trained contextualised word embedding models encode about lexical semantics. The knowledge encoded by word representations about a word's polysemy has also been explored in recent work for static (Jakubowski et al., 2020) and contextualised embeddings (Xypolopoulos et al., 2021; Pimentel et al., 2020). Xypolopoulos et al. (2021) investigate the geometry of ELMo embeddings, and Pimentel et al. (2020) explore the relation between ambiguity and context uncertainty as approximated in the space constructed by multilingual BERT using information-theoretic measures. Both studies find correlations between their polysemy measures and the number of senses in WordNet, whether this information is learnt during pre-training or through exposure to new contexts is unclear. Wiedemann et al. (2019) and Reif et al. (2019) show that BERT can successfully leverage sense annotated data for word sense disambiguation. Aina et al. (2019) probe the hidden representations of a bidirectional (bi-LSTM) LM for lexical and contextual information, and Vulić et al. (2020) investigate the word type-level information encoded in BERT.

Our methodology differs from that in past work. Contrary to Wiedemann et al. (2019) and Reif et al. (2019), we do not use sense annotations to guide the models into establishing sense distinctions, but rather for creating controlled conditions that allow us to analyse BERT's inherent knowledge of lexical polysemy. Vulić et al. (2020) extract type-level representations from these models, whereas we use token-level representations from controlled contexts to infer type-level knowledge relevant to a word's degree of polysemy. The proposed approach relies on the similarity of contextualised representations (Ethayarajh, 2019), which amounts to word usage similarity estimation (Erk et al., 2009). In Chapters 4 and 5 we focused on usage similarity between instance pairs; in this chapter we look at the average usage similarity value for a word and investigate whether it reflects its polysemy.

Our experiments show that representations derived from contextual LMs encode knowledge about words' polysemy acquired through pre-training, which is present in the representations generated for new word instances and is combined with information from these new contexts.

7.2 Polysemy Detection

7.2.1 Dataset Creation

We build our English dataset using SemCor 3.0 (Miller et al., 1993), a corpus manually annotated with WordNet senses (Fellbaum, 1998). It is important to note that we do not use the annotations for training or evaluating any of the models. These only serve to control the composition of the sentence pools that are used for generating contextualised representations, and to analyse the results. We form sentence pools for monosemous (`mono`) and polysemous (`poly`) words that occur at least ten times in SemCor.[1] For each `mono` word, we randomly sample ten of its

[1]We find the number of senses for a word of a specific part of speech (PoS) in WordNet 3.0, which we access through the NLTK interface (Bird et al., 2009).

instances in the corpus. For each `poly` word, we form three sentence pools of size ten reflecting different sense distributions:

- **Balanced** (`poly-bal`). We sample a sentence for each sense of the word in SemCor until a pool of ten sentences is formed.
- **Random** (`poly-rand`). We randomly sample ten `poly` word instances from SemCor. We expect this pool to be highly biased towards a specific sense due to the skewed frequency distribution of word senses (Kilgarriff, 2004; McCarthy et al., 2004). This configuration is closer to the expected natural occurrence of senses in a corpus, it thus serves to estimate the behaviour of the models in a real-world setting.
- **Same sense** (`poly-same`). We sample ten sentences illustrating only one sense of the `poly` word. Although the composition of this pool is similar to that of the `mono` pool (i.e. all instances describe the same sense) we call it `poly-same` because it describes one sense of a polysemous word.[2] Specifically, we want to explore whether BERT representations derived from these instances can serve to distinguish `mono` from `poly` words.

The controlled composition of the `poly` sentence pools allows us to investigate the behaviour of the models when they are exposed to instances of polysemous words describing the same or different senses. There are 1,765 `poly` words in SemCor with at least 10 sentences available.[3] We randomly subsample 418 from these in order to balance the `mono` and `poly` classes. Our English dataset is composed of 836 `mono` and `poly` words, and their instances in 8,195 unique sentences. Table 7.1 shows a sample of the sentences in each pool. For French, Spanish and Greek, we retrieve sentences from the Eurosense corpus (Delli Bovi et al., 2017) which contains texts from Europarl automatically annotated with BabelNet word senses (Navigli and Ponzetto, 2012). We extract sentences from the high precision version[4] of Eurosense, and create sentence pools in the same way as in English, balancing the number of monosemous and polysemous words (418). We determine the number of senses for a word as the number of its Babelnet senses that are mapped to a WordNet sense. This filtering serves to exclude BabelNet senses that correspond to named entities and are not useful for our purposes (such as movie or album titles), and to run these experiments under similar conditions to our English experiments.

7.2.2 Contextualised Word Representations

We experiment with representations generated by three English models: BERT (Devlin et al., 2019)[5], ELMo (Peters et al., 2018a), and context2vec (Melamud et al., 2016). We use the `bert-base-uncased` and `bert-base-cased` models, pre-trained on the BooksCorpus (Zhu et al., 2015) and English Wikipedia. We use 1024-*d* representations from the 5.5B ELMo model,[6]

[2]The polysemous words are the same as in `poly-bal` and `poly-rand`.

[3]We use sentences of up to 100 words.

[4]The high coverage version of Eurosense is larger than the high precision one, but disambiguation is less accurate.

[5]We use Huggingface `transformers` (Wolf et al., 2020)

[6]`https://allennlp.org/elmo`

Setting	Word	Sense	Sentences
mono	hotel.n	INN	The walk ended, inevitably, right in front of his hotel building.
		INN	Maybe he's at the hotel.
poly-same	room.n	CHAMBER	The room vibrated as if a giant hand had rocked it.
		CHAMBER	(...) Tell her to come to Adam's room (...)
poly-bal	room.n	CHAMBER	(...) he left the room, walked down the hall (...)
		SPACE	It gives them room to play and plenty of fresh air.
		OPPORTUNITY	Even here there is room for some variation, for metal surfaces vary (...)

Table 7.1: Example sentences for the monosemous noun *hotel* and the polysemous noun *room*.

and the context representations from a 600-*d* context2vec model pre-trained on the ukWaC corpus (Baroni et al., 2009).[7]

For French, Spanish and Greek, we use BERT models specifically trained for each language:

- `flaubert_base_uncased` (Le et al., 2020) trained on 12.8B tokens from the French WMT19 shared task data (Li et al., 2019), the OPUS collection (Tiedemann, 2012) and Wikipedia, with a 50k BPE vocabulary;
- The BETO model (Cañete et al., 2020) `dccuchile/bert-base-spanish-wwm-uncased` trained on the Spanish parts of Wikipedia and the OPUS Project (Tiedemann, 2012) of a total of 3B tokens, and a 32k vocabulary size;
- Greek BERT `bert-base-greek-uncased-v1` (Koutsikakis et al., 2020), trained on a total of 3.04B tokens coming from the Greek portions of Wikipedia, Europarl (Koehn, 2005) and OSCAR. The vocabulary size is 35k.

We also use the `bert-base-multilingual-cased` model (mBERT) for each of the four languages. mBERT was trained on Wikipedia data of 104 languages.[8] All BERT models generate 768-*d* representations.

7.2.3 The Self-Similarity Metric

All models produce representations that describe word meaning in specific contexts of use. For each instance i of a target word w in a sentence, we extract its representation from: (i) each of the 12 layers of a BERT model;[9] (ii) each of the three ELMo layers; (iii) context2vec. We calculate self-similarity ($SelfSim$) (Ethayarajh, 2019) for w in a sentence pool p and a layer l, by taking the average of the pairwise cosine similarities of the representations of its instances in l:

[7] https://github.com/orenmel/context2vec

[8] The mBERT model developers recommend using the cased version of the model rather than the uncased one, especially for languages with non-Latin alphabets, because it fixes normalisation issues. More details about this model can be found here: https://github.com/google-research/bert/blob/master/multilingual.md.

[9] We also tried different combinations of the last four layers, but this did not improve the results. When a word is split into multiple wordpieces (WPs), we obtain its representation by averaging the WPs.

$$SelfSim_l(w) = \frac{1}{|I|^2 - |I|} \sum_{i \in I} \sum_{\substack{j \in I \\ j \neq i}} cos(x_{wli}, x_{wlj}) \tag{7.1}$$

In formula 7.1, $|I|$ is the number of instances for w (ten in our experiments); x_{wli} and x_{wlj} are the representations for instances i and j of w in layer l. We report the average $SelfSim$ for all w's in a pool p. $SelfSim$ is in the range [-1, 1]. We expect the average $SelfSim$ for monosemous words and words with low polysemy to be higher than that of highly polysemous words. We also expect the `poly-same` pool to have a higher average $SelfSim$ than the other `poly` pools which contain instances of different senses.

Contextualisation has a strong impact on $SelfSim$ since it introduces variation in the token-level representations, making them more dissimilar. The $SelfSim$ value for a word would be 1 with non-contextualised (or static) embeddings, as all its instances would be assigned the same vector. In contextual models, $SelfSim$ is lower in layers where the impact of the context is stronger (Ethayarajh, 2019). It is, however, important to note that contextualisation in BERT models is not monotonic, as shown by previous studies of the models' internal workings (Voita et al., 2019a; Ethayarajh, 2019). Our experiments presented in the next section provide additional evidence in this respect.

7.2.4 Results and Discussion

7.2.4.1 Distinction between `mono` and `poly` Words in English

Figure 7.1 shows the average $SelfSim$ obtained for each sentence pool with representations produced by BERT models. The thin lines in the first plot illustrate the average $SelfSim$ score calculated for `mono` and `poly` words using representations from each layer of the uncased English BERT model. We observe a clear distinction of words according to their polysemy: $SelfSim$ is higher for `mono` than for `poly` words across all layers and sentence pools. BERT establishes a clear distinction even between the `mono` and `poly-same` pools, which contain instances of only one sense. This distinction is important; it suggests that BERT encodes information about a word's monosemous or polysemous nature regardless of the sentences that are used to derive the contextualised representations. BERT produces less similar representations for word instances in the `poly-same` pool compared to `mono`, reflecting that `poly` words can have different meanings.

We also observe a clear ordering of the three `poly` sentence pools: average $SelfSim$ is higher in `poly-same`, which only contains instances of one sense, followed by mid-range values in `poly-rand`, and gets its lowest values in the balanced setting (`poly-bal`). This is noteworthy given that `poly-rand` contains a mix of senses but with a stronger representation of w's most frequent sense than in `poly-bal` (71% vs. 47%).[10]

Our results demonstrate that BERT representations encode two types of lexical semantic knowledge: information about the polysemous nature of words acquired through pre-training

[10]Numbers are macro-averages for words in the pools.

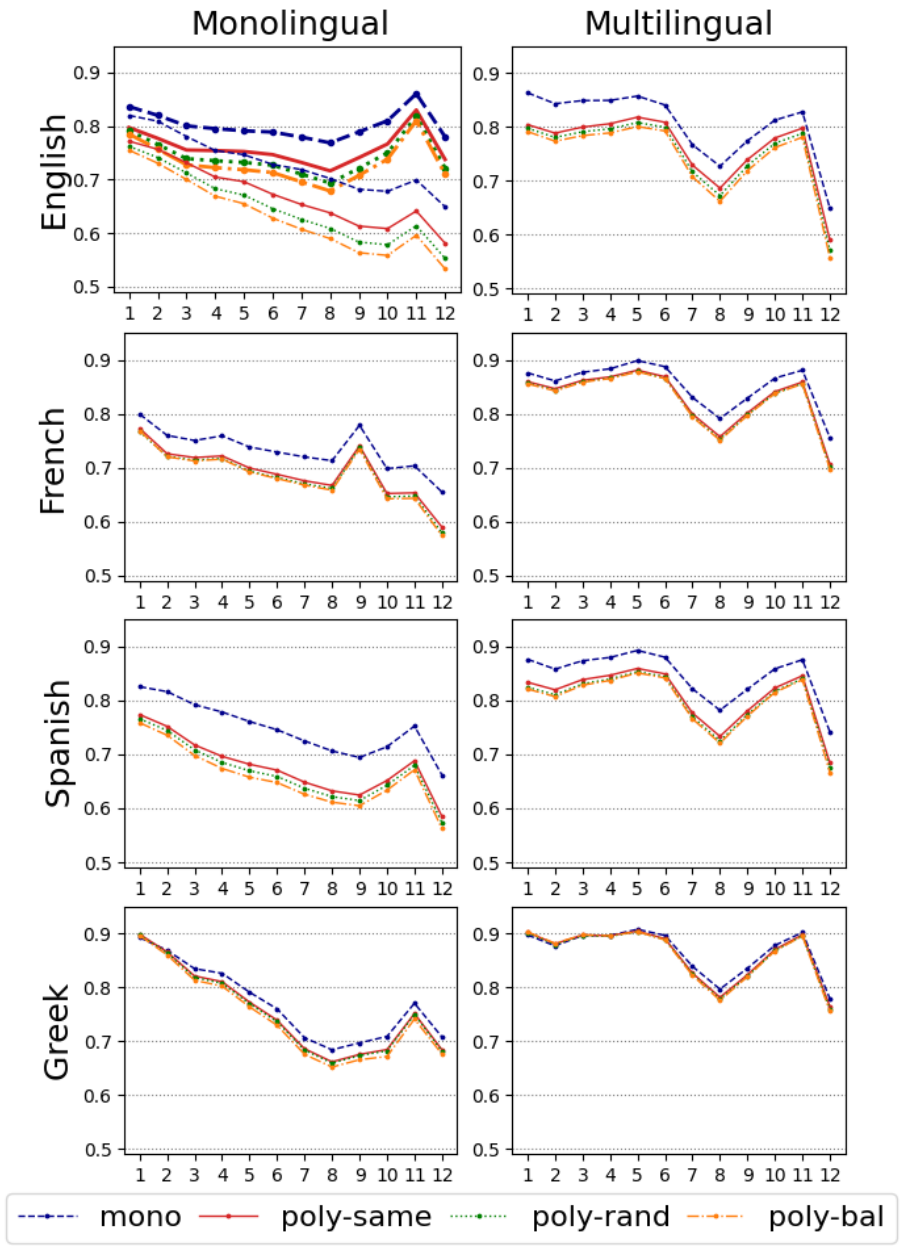


Figure 7.1: Average *SelfSim* obtained with **monolingual BERT** models (left column) and **mBERT** (right column) in all languages across all layers (horizontal axis). In the first plot, thicker lines correspond to the `cased` model.

(as reflected in the distinction between `mono` and `poly-same`) and information from the particular instances of a word used to create the contextualised representations (as shown by the finer-grained distinctions between different `poly` settings). BERT's knowledge about polysemy can be due to differences in the types of context where words of different polysemy levels are used. We expect `poly` words to be seen in more varied contexts than `mono` words, reflecting their different senses. BERT encodes this variation with the LM objective through exposure to large amounts of data, and this is reflected in the representations. The same ordering pattern is observed with mBERT (right column of Figure 7.1), with ELMo (Figure 7.2) and context2vec

(left part of Table 7.2). This suggests that these models also have some inherent knowledge about lexical polysemy, but differences are less clearly marked than in BERT.

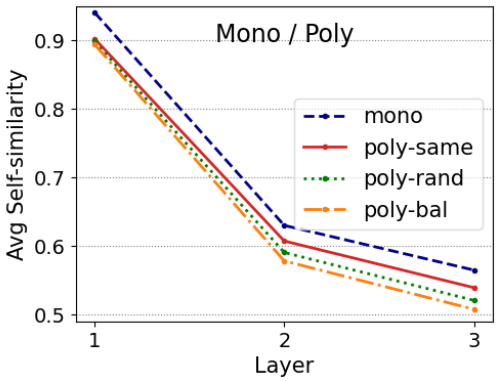


Figure 7.2: Comparison of **ELMo** average *SelfSim* for `mono` and `poly` lemmas.

mono/poly		poly bands			
			poly-same	poly-rand	poly-bal
mono	0.400	mono		0.400	
poly-same	0.385	low	0.382	0.368	0.362
poly-rand	0.375	mid	0.381	0.358	0.349
poly-bal	0.353	high	0.386	0.356	0.338

Table 7.2: Average *SelfSim* obtained with context2vec for words in different sentence pools. The first two columns of the table show the average *SelfSim* for `mono` and `poly` words. These results are presented in Section 7.2. The other columns show the average *SelfSim* obtained for `poly` words in different polysemy bands (described in Section 7.3).

Using the `cased` model leads to an overall increase in *SelfSim* and to smaller differences between bands, as shown by the thick lines in the first plot of Figure 7.1. Our explanation for the lower distinction ability of the `bert-base-cased` model is that it encodes sparser information about words than the `uncased` model. It was trained on a more diverse set of strings, so many WPs are present in both their capitalised and non-capitalised form in the vocabulary. In spite of that, it has a smaller vocabulary size (29K WPs) than the uncased model (30.5K). Also, a higher number of WPs correspond to word parts than in the uncased model (6,478 vs 5,829).

We test the statistical significance of the `mono`/`poly-rand` distinction using unpaired two-samples t-tests when the normality assumption is met (as determined with Shapiro Wilk's tests). Otherwise, we run a Mann Whitney U test, the non-parametrical alternative of this t-test. In order to lower the probability of type I errors (false positives) that increases when performing multiple tests, we correct p-values using the Benjamini–Hochberg False Discovery Rate (FDR) adjustment (Benjamini and Hochberg, 1995). Our results show that differences are significant across all embedding types and layers ($\alpha = 0.01$).

The decreasing trend in *SelfSim* observed for BERT in Figure 7.1, and the peak in layer 11, confirm the phases of context encoding and token reconstruction observed by Voita et al.

	Model	Avg $SelfSim$(mono) − Avg $SelfSim$(poly-rand)
EN	bert-base-uncased	0.10_{10}
	bert-base-cased	0.08_{8}
	mBERT	0.08_{12}
	ELMo	0.04_{3}
	context2vec	0.03
FR	Flaubert	0.08_{12}
	mBERT	0.05_{12}
ES	BETO	0.09_{4}
	mBERT	0.07_{12}
EL	GreekBERT	0.03_{10}
	mBERT	0.02_{12}

Table 7.3: Largest difference in $SelfSim$ between mono and poly-rand for all models. Subscripts indicate the model layer.

(2019a). In earlier layers, context variation makes representations more dissimilar and $SelfSim$ decreases. In the last layers, information about the input token is recovered for LM prediction and similarity scores are boosted.

Our results show clear distinctions across all BERT and ELMo layers. This suggests that lexical information is spread throughout the layers of the models, and contributes new evidence to the discussion on the localisation of semantic information (Rogers et al., 2020; Vulić et al., 2020).

7.2.4.2 Distinction between mono and poly Words in Other Languages

The left column of Figure 7.1 also shows the average $SelfSim$ obtained for French, Spanish and Greek words using monolingual models. Flaubert, BETO and Greek BERT representations clearly distinguish mono and poly words, but average $SelfSim$ values for different poly pools are much closer than in English. BETO seems to capture these fine-grained distinctions slightly better than the French and Greek models. The right column of the Figure shows results obtained with mBERT representations. We observe the highly similar average $SelfSim$ values assigned to different poly pools, which show that distinction is harder than in monolingual models.

Statistical tests show that the difference between $SelfSim$ values in mono and poly-rand is significant in all layers of BETO, Flaubert, Greek BERT, and mBERT for Spanish and French.[11] Table 7.3 shows the biggest difference in $SelfSim$ between mono and poly-rand per model. The magnitude of the difference in Greek BERT is smaller compared to the other monolingual BERT models (0.03 vs. 0.09 in BETO).

[11] In mBERT for Greek, the difference is significant in ten layers.

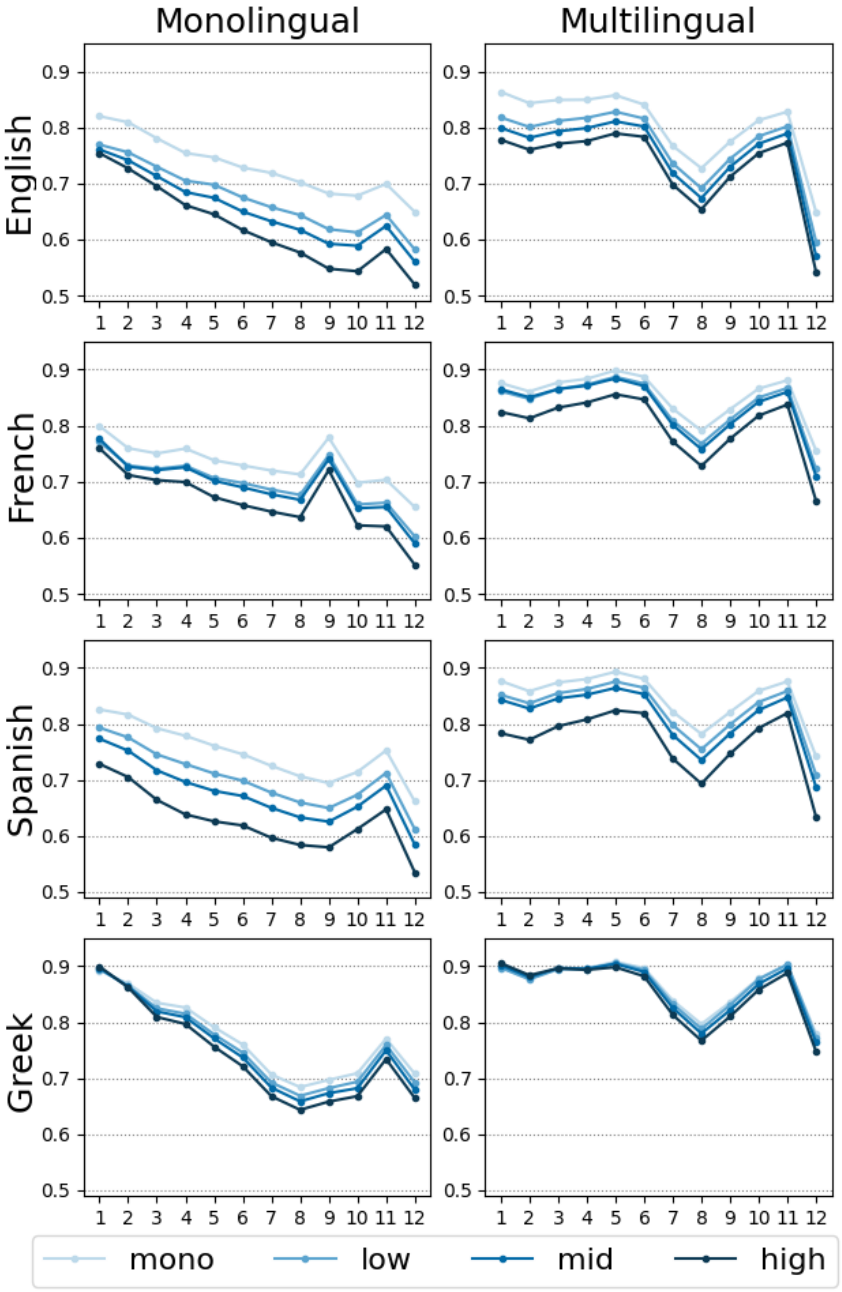


Figure 7.3: Average *SelfSim* obtained with **monolingual BERT** models (left column) and **mBERT** (right column) in all languages for `mono` lemmas and `poly` lemmas in different polysemy bands in the `poly-rand` sentence pool.

7.3 Polysemy Level Prediction

7.3.1 *SelfSim*-based Ranking

In this set of experiments, we explore the impact of words' degree of polysemy on the representations. We control for this factor by grouping words into three polysemy bands, as in McCarthy et al. (2016), which correspond to a specific number of senses (k): `low`: $2 \leq k \leq 3$, `mid`: $4 \leq k \leq$

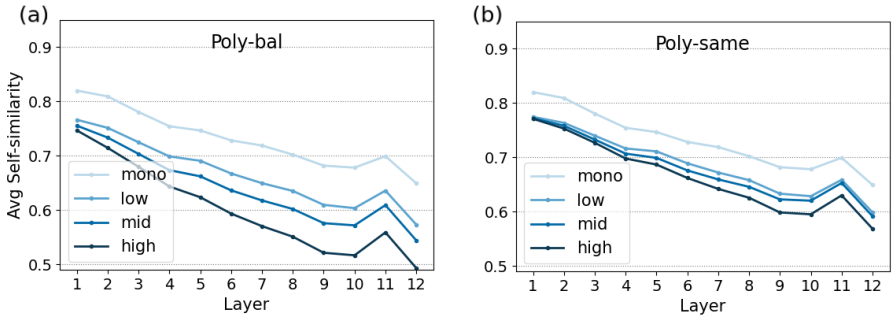


Figure 7.4: Comparison of **BERT** average *SelfSim* for `mono` and `poly` lemmas in different polysemy bands in the English `poly-same` and `poly-bal` sentence pools.

6, `high`: $k > 6$. For English, the three bands are populated with a different number of words: `low`: 551, `mid`: 663, `high`: 551. In the other languages, we form bands containing 300 words each.[12] In Figure 7.3, we compare `mono` words with lemmas in each polysemy band, in terms of average *SelfSim*. Values for `mono` words are taken from Section 7.2. For `poly` words, we use representations from the `poly-rand` sentence pool which better approximates natural word occurrence in a corpus. For comparison, we report results obtained in English using sentences from the `poly-same` and `poly-bal` pools in Figure 7.4. We include the plots for `poly-bal` and `poly-same` for the other models in Appendix A.3.1.

In English, the pattern is clear in all plots: *SelfSim* is higher for `mono` than for `poly` words in any band, confirming that BERT is able to distinguish `mono` from `poly` words at different polysemy levels. The range of *SelfSim* values for a band is inversely proportional to its k: words in `low` get higher values than words in `high`. The results denote that the meaning of highly polysemous words is more variable (lower *SelfSim*) than the meaning of words with fewer senses. As expected, scores are higher and inter-band similarities are closer in `poly-same` (cf. Figure 7.4 (b)) compared to `poly-bal` and `poly-rand`, where distinctions are clearer. The observed differences confirm that BERT can predict the polysemy level of words, even from instances describing the same sense.

We observe similar patterns with ELMo (cf. Figure 7.5) and context2vec representations in `poly-rand` (right part of Table 7.2) but smaller absolute inter-band differences. In `poly-same`, both models fail to correctly order the bands. Overall, our results highlight that BERT encodes higher quality knowledge about polysemy. We test the significance of the inter-band differences in two subsequent polysemy bands (`mono`→`low`, `low`→`mid`, `mid`→`high`) detected in `poly-rand` using the same approach as in Section 7.2.4.1. These are significant in all but a few[13] layers of the models.

The bands are also correctly ranked in the other three languages, but with smaller inter-band differences than in English, especially in Greek where clear distinctions are only made in a few middle layers. This variation across languages can be explained to some extent by the quality of the automatic EuroSense annotations, which has a direct impact on the quality

[12] We only used 418 of these polysemous words in Section 7.2 in order to have balanced `mono` and `poly` pools.
[13] `low`→`mid` in ELMo's third layer, and `mid`→`high` in context2vec and in BERT's first layer.

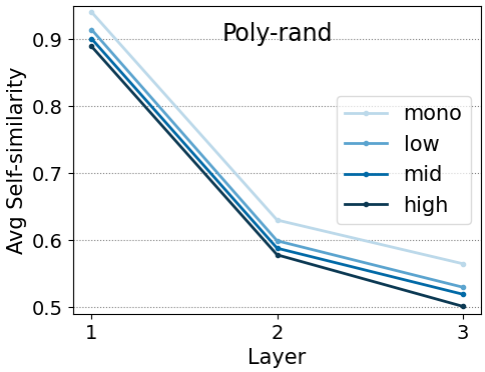


Figure 7.5: Comparison of ELMo average $SelfSim$ for `mono` lemmas and `poly` lemmas in different polysemy bands in the `poly-rand` sentence pool.

of the sentence pools. Results of a manual evaluation conducted by Delli Bovi et al. (2017) showed that WSD precision is ten points higher in English (81.5) and Spanish (82.5) than in French (71.8). The Greek portion, however, has not been evaluated.

Plots in the right column of Figure 7.3 show results obtained using mBERT. Similarly to the previous experiment (Section 7.2.4), mBERT overall makes less clear distinctions than the monolingual models. The `low` and `mid` bands often get similar $SelfSim$ values, which are close to `mono` in French and Greek. Still, inter-band differences are significant in most layers of mBERT and the monolingual French, Spanish and Greek models.[14]

7.3.2 Anisotropy Analysis

In order to better understand the reasons behind the smaller inter-band differences observed with mBERT, we conduct an additional analysis of the models' anisotropy. We create 2,183 random word pairs from the English `mono`, `low`, `mid` and `high` bands, and 1,318 in each of the other languages.[15] We calculate the cosine similarity between two random instances of the words in each pair and take the average over all pairs ($RandSim$). The plots in the left column of Figure 7.6 show the results. We observe a clear difference in the scores obtained by monolingual models (solid lines) and mBERT (dashed lines). Clearly, mBERT assigns higher similarities to random words, an indication that its semantic space is more anisotropic than the one built by monolingual models. High anisotropy means that representations occupy a narrow cone in the vector space, which results in lower quality similarity estimates and in a model's limited potential to establish clear semantic distinctions.

We also compare $RandSim$ to the average $SelfSim$ obtained for `poly` words in the `poly-rand` sentence pool (Section 7.2). In a quality semantic space, we would expect $SelfSim$ (between same word instances) to be much higher than $RandSim$. The right column of Figure 7.6 shows the difference between these two scores. $diff$ in a layer l is calculated as in Equation 7.2:

$$diff_l = \text{Avg}SelfSim_l(\texttt{poly}-\texttt{rand}) - RandSim_l \tag{7.2}$$

[14] With the exception of `mono`→`low` in mBERT for Greek and `low`→`mid` in Flaubert and mBERT for French.
[15] 1,318 is the total number of words across bands in French, Spanish and Greek.

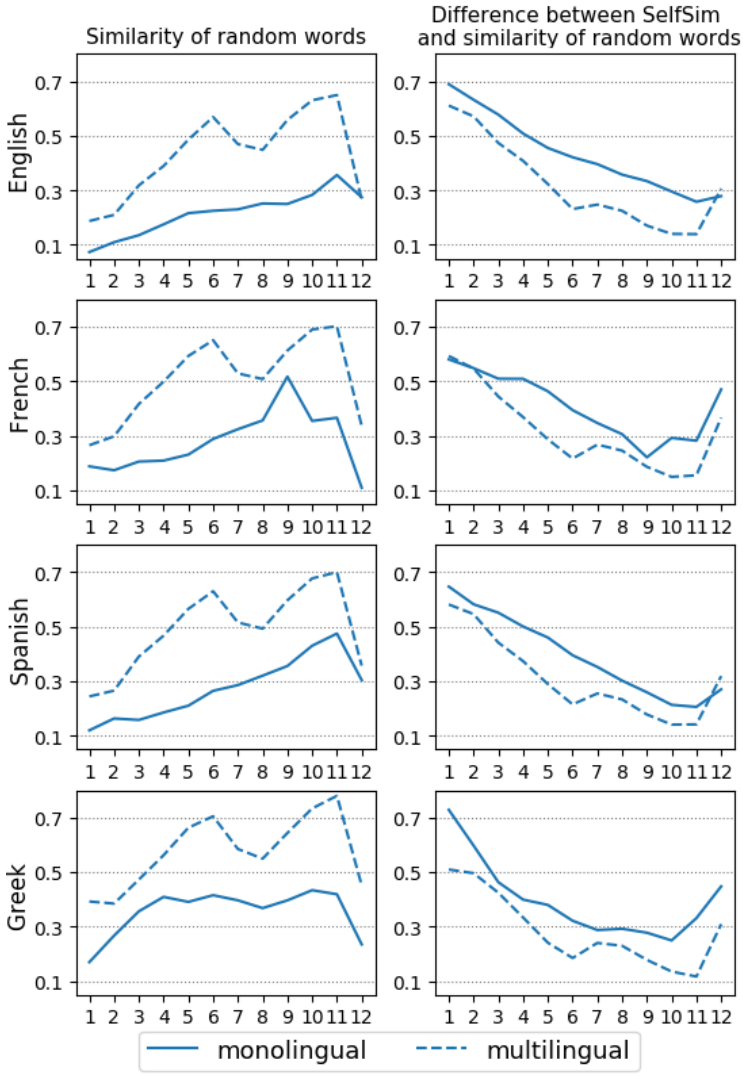


Figure 7.6: The left plots show the similarity between random words in the models for each language. Plots on the right show the difference between the similarity between random words (*RandSim*) and *SelfSim* of `poly-rand`.

We observe that the difference is smaller in the space built by mBERT, which is more anisotropic than monolingual spaces, and becomes very low in the last layers of the model. This result confirms the lower quality of mBERT's semantic space compared to monolingual models.

Finally, we believe that another factor behind the worse mBERT results is that the multilingual WP vocabulary is mostly English-driven, resulting in arbitrary partitionings of words in the other languages. This word splitting procedure must have an impact on the quality of the lexical information in mBERT representations.

7.4 Analysis by Frequency and PoS

Given the strong correlation between word frequency and number of senses (Zipf, 1945), we explore the impact of frequency on BERT representations. Our goal is to determine the extent

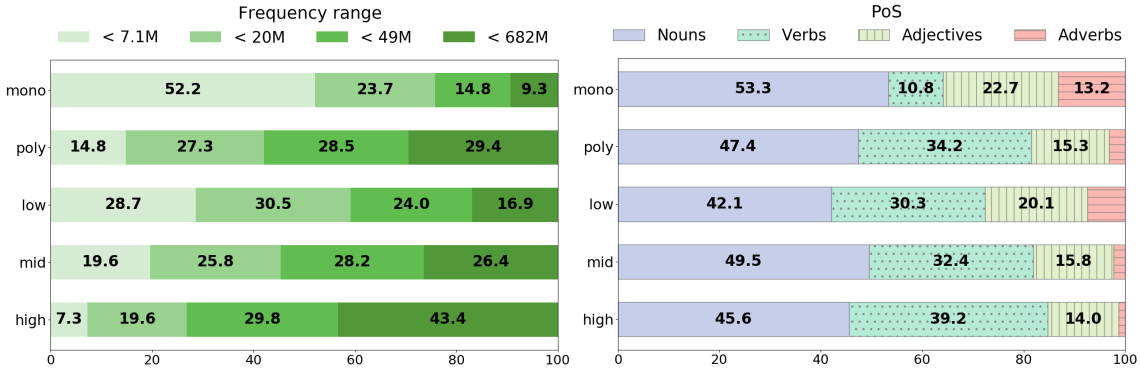


Figure 7.7: Composition of the English word bands in terms of frequency (left) and grammatical category (right).

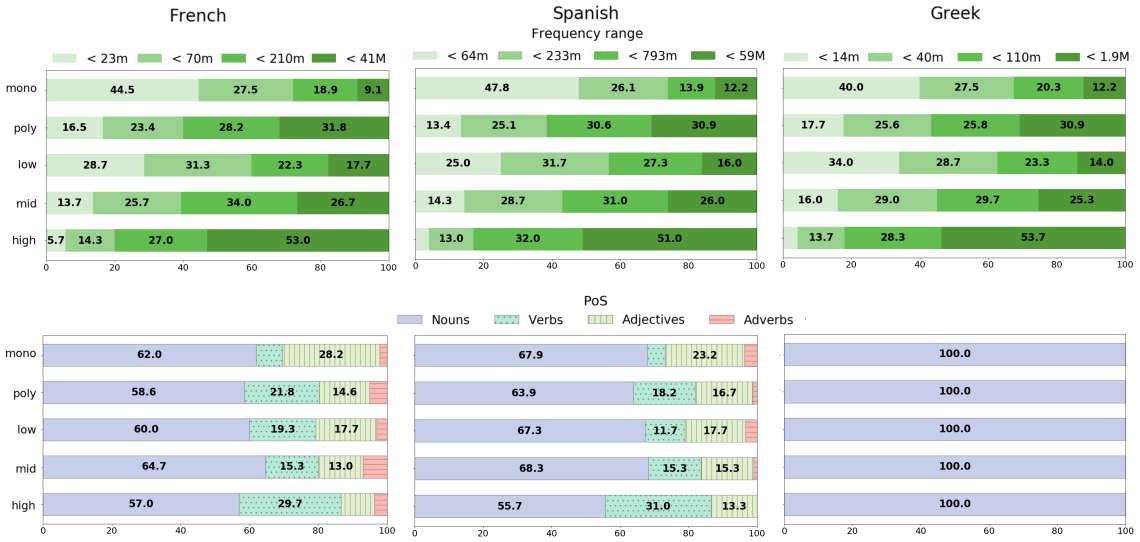


Figure 7.8: Composition of the French, Spanish and Greek word bands in terms of frequency (top) and grammatical category (bottom).

to which it influences the good `mono/poly` detection results obtained in Sections 7.2.4 and 7.3.1. Similarly, we investigate the impact of part of speech (PoS) categories on representations, as it is also related to polysemy.

7.4.1 Dataset Composition

We perform this analysis in English using frequency information from Google Ngrams (Brants and Franz, 2006). For French, Spanish and Greek, we use frequency counts gathered from the OSCAR corpus (Suárez et al., 2019). We split the words into four ranges (F) corresponding to the quartiles of frequencies in each dataset. Each range f in F contains the same number of words. We provide detailed information about the composition of the English dataset in Figure 7.7.[16] Figure 7.7 (left) shows that `mono` words are much less frequent than `poly` words. Figure 7.7 (right) shows the distribution of different PoS categories in each band. Nouns are

[16]The composition of each band is the same as in Sections 7.2 and 7.3.

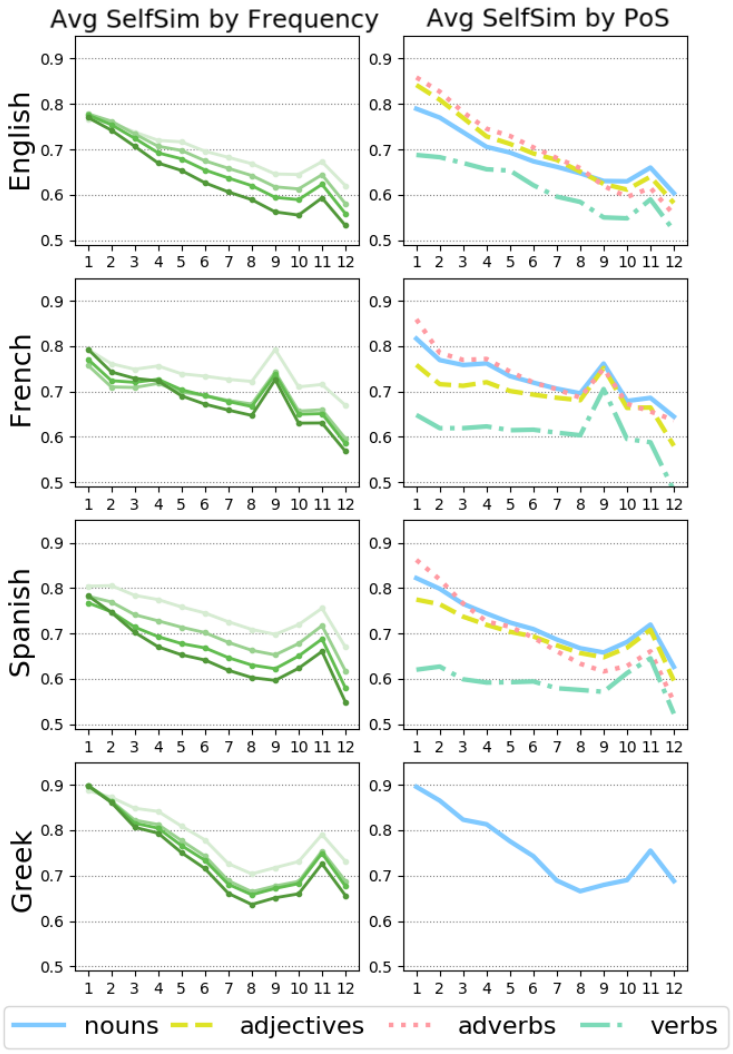


Figure 7.9: Average *SelfSim* obtained for words of different frequencies and part of speech categories with **monolingual BERT** representations in different languages, using the `poly-rand` sentence pool. The frequency ranges used for each language are the same as in Figures 7.7 and 7.8, where a darker colour indicates a higher frequency range.

the prevalent category in all bands and verbs are less present among `mono` words (10.8%), as expected. Finally, adverbs are hardly represented in the `high` polysemy band (1.2% of all words). The composition of the bands in the other languages is shown in Figure 7.8. We observe the same tendencies as in English, except for PoS in the Greek dataset, because all sense-annotated Greek words in EuroSense are nouns.

7.4.2 Self-Sim by Frequency Range and PoS Category

We examine the average BERT *SelfSim* per frequency range in `poly-rand` (Figure 7.9, left column). We carry out this analysis for the monolingual BERT models in all languages. The clear ordering by range suggests that BERT can successfully distinguish words by their frequency, especially in the last layers. Plots in the right column of Figure 7.9 show the average *SelfSim*

POS-bal

	Nouns	**Verbs**	**Adjectives**	**Adverbs**
en	198	45	64	7
fr	171	32	29	9
es	167	22	40	0

FREQ-bal

en	**7.1**M	**20**M	**49**M	**682**M
	40	99	62	39
fr	**23**m	**70**m	**210**m	**41**M
	17	43	67	38
es	**64**m	**233**m	**793**m	**59**M
	12	39	58	48
el	**14**m	**40**m	**111**m	**1.9**M
	13	41	70	42

Table 7.4: Content of the polysemy bands in the POS-bal and FREQ-bal settings. All bands for a language contain the same number of words of a specific grammatical category or frequency range. M stands for a million and m for a thousand occurrences of a word in a corpus.

for words of each PoS category. Verbs have the lowest $SelfSim$ which is not surprising given that they are highly polysemous (as shown in Figures 7.7 and 7.8). We observe similar trends in all languages.

7.4.3 Controlling for Frequency and PoS

We conduct an additional experiment where we control for the composition of the `poly` bands in terms of grammatical category and word frequency. We call these two settings POS-bal and FREQ-bal. We define n_{pos}, the smallest number of words of a specific PoS that can be found in a band. We form the POS-bal bands by subsampling from each band the same number of words (n_{pos}) of that PoS. For example, all POS-bal bands have n_n nouns and n_v verbs. We follow a similar procedure to balance the bands by frequency in the FREQ-bal setting. In this case, n_f is the minimum number of words of a specific frequency range f that can be found in a band. We form the FREQ-bal dataset by subsampling from each band the same number of words (n_f) of a given range f in F.

Table 7.4 shows the distribution of words per PoS and frequency range in the POS-bal and FREQ-bal bands for each language. The table reads as follows: the English POS-bal bands contain 198 nouns, 45 verbs, 64 adjectives and 7 adverbs; similarly for the other two languages. In FREQ-bal, each English band contains 40 words that occur less than 7.1M times in Google Ngrams, 99 words that occur between 7.1M and 20M times, and so on and so forth.

We examine the average $SelfSim$ values obtained for words in each band in `poly-rand`. Figure 7.10 shows the results for monolingual BERT models. We observe that the `mono` and `poly` words in the POS-bal and FREQ-bal bands are ranked similarly to Figure 7.3. This shows

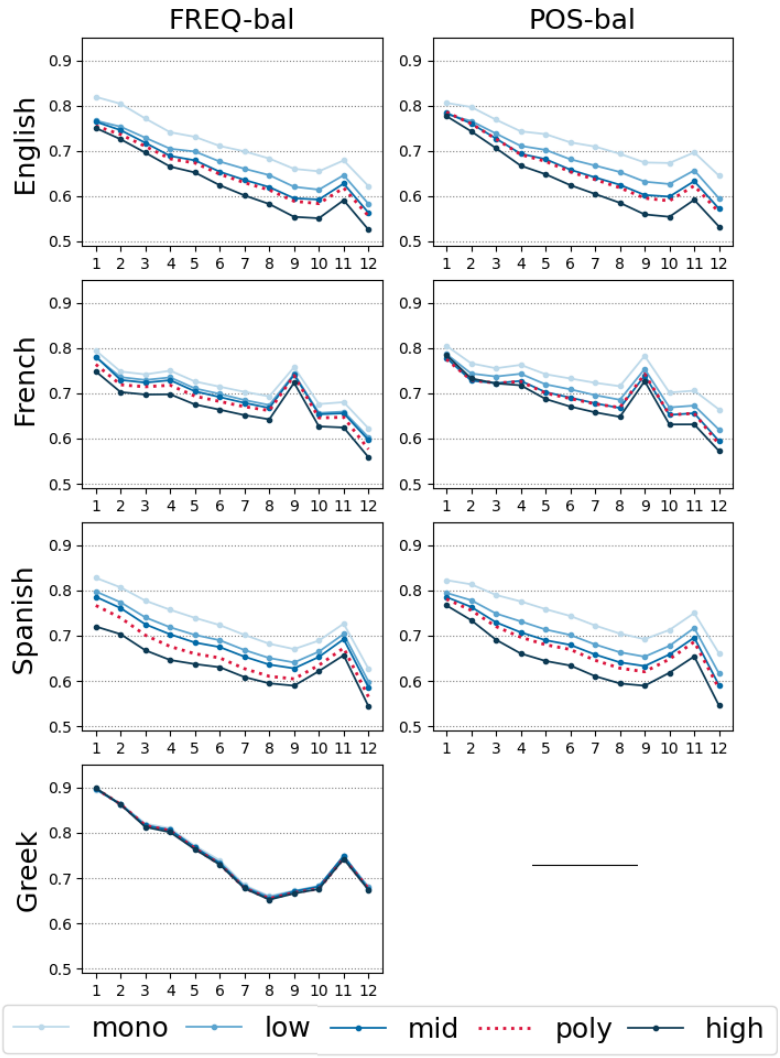


Figure 7.10: Average *SelfSim* inside the `poly` bands balanced for frequency (FREQ-bal) and part of speech (POS-bal). *SelfSim* is calculated using representations generated by **monolingual BERT** models from sentences in each language-specific pool. We do not balance the Greek dataset for PoS because it only contains nouns.

that BERT's polysemy predictions do not rely on frequency or part of speech. The only exception is Greek BERT which cannot establish correct inter-band distinctions when the influence of frequency is neutralised in the FREQ-bal setting. A general observation that applies to all models is that although inter-band distinctions become less clear, the ordering of the bands is preserved. We observe the same trend with ELMo (Figure 7.11) and context2vec (Table 7.5). Results with mBERT are included in Appendix A.3.2.

Statistical tests show that all inter-band distinctions established by English BERT are still significant in most layers of the model.[17] This is not the case for ELMo and context2vec, which can distinguish between `mono` and `poly` words but fail to establish significant distinctions

[17]Note that the sample size in this analysis is smaller compared to that used in Sections 7.2.4 and 7.3.1.

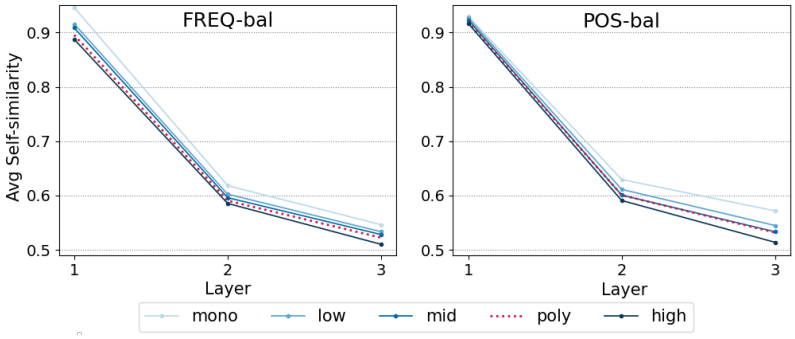


Figure 7.11: Average *SelfSim* inside the `poly` bands balanced for frequency (FREQ-bal) and part of speech (POS-bal), calculated using representations from the **ELMo** model.

	poly-rand	
	FREQ-bal	POS-bal
mono	0.389	0.396
poly	0.367	0.370
low	0.368	0.372
mid	0.363	0.364
high	0.359	0.361

Table 7.5: Average *SelfSim* obtained with **context2vec** in the FREQ-bal and POS-bal bands from the `poly-rand` sentence pool.

between polysemy bands in the balanced settings.[18] For French and Spanish, the statistical analysis shows that all distinctions in POS-bal are significant in at least one layer of the models. The same applies to the `mono`→`poly` distinction in FREQ-bal but finer-grained distinctions get lost, also in Greek mBERT.[19]

7.5 Classification by Polysemy Level

Our finding that word instance similarity differs across polysemy bands suggests that this feature can be useful for classification. In this Section, we probe the representations for polysemy using a classification experiment where we test their ability to guess whether a word is polysemous, and which `poly` band it falls in. We use the `poly-rand` sentence pools and a standard train/dev/test split (70/15/15%) of the data. For the `mono`/`poly` distinction (i.e. the data used in Section 7.2), this results in 584/126/126 words per subset in each language. To guarantee a fair evaluation, we make sure there is no overlap between the words in the three sets. We use two types of features: (i) the *SelfSim* for a word; (ii) all pairwise cosine similarities

[18]Interestingly, ELMo's first layer, which is character-based, made a significant distinction between `mono` and `poly` words in Section 7.2. This is due to the fact that, in English, verbs (which are more prevalent in `poly` than in `mono`) can be found in more different forms than other parts of speech. When removing the effect of PoS in POS-bal, this distinction in the first layer is lost.

[19]With a few exceptions: for example, `mono`→`low` and `mid`→`high` are significant in all BETO layers.

		mono/poly		poly bands	
	Model	*SelfSim*	*pairCos*	*SelfSim*	*pairCos*
EN	BERT	0.76_{10}	$\mathbf{0.79}_{8}$	$\mathbf{0.49}_{10}$	0.46_{10}
	mBERT	0.77_{8}	0.75_{8}	0.46_{12}	$.43_{12}$
	ELMo	0.69_{2}	0.63_{3}	0.37_{2}	0.34_{3}
	context2vec	0.61	0.61	0.34	0.31
	Frequency	0.77		0.41	
FR	Flaubert	0.58_{7}	0.55_{6}	0.29_{8}	0.27_{9}
	mBERT	$\mathbf{0.66}_{9}$	0.64_{9}	$\mathbf{0.38}_{7}$	$\mathbf{0.38}_{8}$
	Frequency	0.61		0.37	
ES	BETO	$\mathbf{0.70}_{9}$	0.66_{7}	0.42_{6}	$\mathbf{0.48}_{5}$
	mBERT	0.69_{11}	0.64_{7}	0.38_{9}	0.43_{7}
	Frequency	0.67		0.41	
EL	GreekBERT	$\mathbf{0.70}_{4}$	0.64_{4}	0.34_{4}	$\mathbf{0.38}_{6}$
	mBERT	0.60_{7}	0.65_{7}	0.32_{11}	0.34_{9}
	Frequency	0.63		0.35	
	Baseline	0.50		0.25	

Table 7.6: Accuracy of binary (`mono/poly`) and multi-class (`poly` bands) classifiers using *SelfSim* and *pairCos* features on the test sets. Comparison to a baseline that predicts always the same class and a classifier that only uses log frequency as feature. Subscripts denote the layers used.

collected for its instances, which results in 45 features per word (*pairCos*). We train a binary logistic regression classifier for each type of representation and feature.

As explained in Section 7.3, the three `poly` bands (`low`, `mid` and `high`) and `mono` contain a different number of lemmas. For classification into polysemy bands, we balance each class by randomly subsampling words from each band. In total, we use 1,168 words for training, 252 for development and 252 for testing (70/15/15%) in English. In the other languages, we use a split of 840/180/180 words. We train multi-class logistic regression classifiers with the two types of features, *SelfSim* and *pairCos*. We compare the results of the classifiers to a baseline that predicts always the same class, and to a frequency-based classifier which only uses the words' log frequency in Google Ngrams, or in the OSCAR corpus, as a feature.

Table 7.6 presents the classification accuracy on the test set. We report results obtained with the best layer for each representation type and feature as determined on the development sets. In English, best accuracy is obtained by BERT in both the binary (0.79) and multiclass settings (0.49), followed by mBERT (0.77 and 0.46). Despite its simplicity, the frequency-based classifier obtains better results than context2vec and ELMo, and performs on par with mBERT in the binary setting (0.77). This confirms that frequency information is highly relevant for the `mono-poly` distinction. All classifiers outperform the same class baseline. These results are very encouraging, showing that BERT embeddings can be used to determine whether a word has multiple meanings, and provide a rough indication of its polysemy level. Results in the other three languages are not as high as those obtained in English, but most models give

higher results than the frequency-based classifier.[20]

7.6 Conclusion

We analysed the similarity estimates derived from different types of contextualised representations, searching for information about words' polysemy level. We found that English BERT representations encode rich information about lexical polysemy. Our experimental results suggest that this high quality knowledge about words, which allows BERT to detect polysemy in different configurations and across multiple layers, is acquired during pre-training, as it is present in BERT representations regardless of the contexts used to derive them. This is an important finding, which shows that exposure to large amounts of data with the MLM pre-training objective allows BERT to capture this property of words. Our findings hold for the English BERT as well as for BERT models in other languages, as shown by our experiments on French and Spanish, and to a lesser extent for Greek BERT, multilingual BERT, context2vec and ELMo.

We can envisage various theoretical and application-related extensions for this work. The polysemy knowledge revealed by the models can serve to develop novel methodologies for improved cross-lingual alignment of embedding spaces and cross-lingual transfer (Artetxe et al., 2017; Smith et al., 2017), pointing to less polysemous words that can serve as stable anchors. Predicting the polysemy level of words can also be useful for determining the context needed for acquiring representations that properly reflect the meaning of word instances in running text. From a more theoretical standpoint, this work can be useful for studying the organisation of the semantic space in different languages and also for detecting lexical semantic change (Giulianelli et al., 2020; Martinc et al., 2020) .

[20] Only exceptions are Greek mBERT in the multi-class setting, and Flaubert in both settings.

Chapter 8

Scalar Adjective Identification and Ranking

8.1 Introduction

In previous chapters of the thesis we have mainly addressed aspects of meaning related to lexical ambiguity. We now shift our focus and investigate word relationships, rather than the internal semantic properties of words. In this chapter, we specifically explore the intensity relationship between scalar adjectives.

Scalar adjectives describe a property of a noun at different degrees of intensity. Identifying the scalar relationship that exists between their meaning (for example, the increasing intensity between *pretty*, *beautiful* and *gorgeous*) is useful for text understanding, for both humans and automatic systems. It can serve to define the sentiment and subjectivity of a text, perform inference and textual entailment (Van Tiel et al., 2016; McNally, 2016) (*wonderful* $\rightarrow$ *good* but *good* $\nrightarrow$ *wonderful*), build question answering and recommendation systems (de Marneffe et al., 2010), and assist language learners in distinguishing between semantically similar words (Sheinman and Tokunaga, 2009).

In this chapter, we investigate the knowledge that BERT representations encode about the intensity of scalar adjectives, and propose methodology for estimating it. Given that this property is acquired by humans during language learning, we expect a language model (LM) exposed to massive amounts of text data during training to have also acquired some notion of adjective intensity. We explore this hypothesis using representations extracted from different layers of this deep neural model. We propose a method inspired by gender bias work (Bolukbasi et al., 2016) for detecting the intensity relationship of two adjectives. We view intensity as a direction in the semantic space which, once identified, can serve to determine the intensity of new adjectives. We evaluate the representations generated by BERT against gold standard adjective scales ordered by intensity (de Melo and Bansal, 2013; Wilkinson and Oates, 2016; Cocos et al., 2018) and apply them directly to a question answering task (de Marneffe et al., 2010). Our results show that BERT clearly encodes the intensity variation between adjectives on scales describing different properties. We also propose to extend scalar adjective ranking

to new languages (Section 8.3). Previous research has focused on English, mainly due to the availability of datasets for evaluation. In order to promote scalar adjective research in new languages, we introduce new scalar adjective datasets in French, Spanish and Greek and use our resource-lean method with monolingual and multilingual contextual models.

Not all adjectives, however, express intensity or degree. Relational adjectives are derived from nouns (e.g. *wood* → *wooden*, *chemistry* → *chemical*), have no antonyms and serve to classify a noun (McNally and Boleda, 2004). Distinguishing between scalar and relational adjectives is important: it allows to identify words that can serve to assess the emotional tone of a given text, as opposed to words that mostly contribute to its content. This distinction is relevant for Sentiment Analysis and recommendation systems. We introduce a new binary classification task for scalar adjective identification (Section 8.4) which examines the models' capability to identify scalar adjectives. We probe contextualised representations and report baseline results for future comparison on this task.

The analysis of scalar adjective relationships in the literature has often been decomposed into two steps: grouping related adjectives together and ranking adjectives in the same group according to intensity. The first step can be performed by distributional clustering approaches (Hatzivassiloglou and McKeown, 1993; Pang et al., 2008) which can also address adjectival polysemy. *Hot*, for example, can be on the TEMPERATURE scale (a *warm* → *hot* → *scalding* drink), the ATTRACTIVENESS (a *pretty* → *hot* → *sexy* person) or the INTEREST scale (an *interesting* → *hot* topic), depending on the attribute it modifies. Other works (Sheinman and Tokunaga, 2009; de Melo and Bansal, 2013; Wilkinson, 2017) directly address the second step, ranking groups of semantically related adjectives from lexicographic resources (e.g. WordNet) (Fellbaum, 1998). We focus on the ranking step.

Adjective ranking has traditionally been performed using pattern-based approaches which extract lexical or syntactic patterns indicative of an intensity relationship from large corpora (Sheinman and Tokunaga, 2009; de Melo and Bansal, 2013; Sheinman et al., 2013; Shivade et al., 2015). For example, the patterns "*X*, but not *Y*" and "not just *X* but *Y*" provide evidence that X is an adjective less intense than Y (e.g. "*cold*, but not *freezing*"). Another common approach is lexicon-based and draws upon a resource that maps adjectives to scores encoding sentiment polarity (positive or negative) and intensity. Such resources can be manually created, like the SO-CAL lexicon (Taboada et al., 2011), or automatically compiled by mining adjective orderings from star-valued product reviews where people's comments have associated ratings (de Marneffe et al., 2010; Rill et al., 2012; Sharma et al., 2015; Ruppenhofer et al., 2014). Cocos et al. (2018) combine knowledge from lexico-syntactic patterns and the SO-CAL lexicon with evidence from paraphrases in the Paraphrase Database (PPDB) (Ganitkevitch et al., 2013; Pavlick et al., 2015). For example, if "very *X*" is a paraphrase of "*Y*" (e.g. "very *cold*" = "*freezing*"), this is an indication that X is of lower intensity than Y.

Our approach to scalar adjective ranking is novel in that it does not need specified patterns or access to lexicographic resources. It, instead, relies on the knowledge about intensity encoded in scalar adjectives' contextualised representations. Our best performing method is inspired by work on gender bias which relies on simple vector arithmetic to uncover gender-

related stereotypes. In gender bias work, a gender direction is determined (for example, by comparing the embeddings of *she* and *he*, or *woman* and *man*) and the projection of the vector of a potentially biased word on this direction is then calculated (Bolukbasi et al., 2016; Zhao et al., 2018).

Kim and de Marneffe (2013) also consider vector distance in the semantic space to encode scalar relationships between adjectives. They examine a small set of word pairs, and observe that the middle point in space between the static embeddings of two antonyms (e.g. *furious* and *happy*) falls close to the embedding of a mid-ranked word in their scale (e.g. *unhappy*). Their experiments rely on antonym pairs extracted from WordNet. We show that contextualised representations are a better fit for this task than static embeddings, encoding rich information about adjectives' meaning and intensity. Our work contributes towards the study of the knowledge pre-trained LMs encode about word meaning.

8.2 English Scalar Adjective Ranking

8.2.1 Data

We experiment with three scalar adjective datasets.

DEMELO (de Melo and Bansal, 2013).[1] Adjective sets were extracted from WordNet 'dumbbell' structures (Gross and Miller, 1990), starting with antonym pairs as the poles and extracting adjectives that are similar to each of the antonyms. The sets thus represented full scales (e.g. from *horrible* to *awesome*), which were partitioned into half-scales (from *horrible* to *bad*, and from *good* to *awesome*) based on pattern-based evidence in the Google N-Grams corpus (Brants and Franz, 2006). Half-scales contain near-synonyms that only differ in intensity. The dataset contains 87 half-scales with 548 adjective pairs, manually annotated for intensity relations (<, >, and =).

CROWD (Cocos et al., 2018).[2] The dataset consists of a set of adjective scales with high coverage of the PPDB vocabulary. It was constructed by a three-step process: crowd workers were first asked to determine whether pairs of adjectives describe the same attribute (e.g. TEMPERATURE) and should, therefore, belong to the same scale. Sets of same-scale adjectives were then refined over multiple rounds. Finally, workers ranked the adjectives in each set by intensity. The final dataset includes 330 adjective pairs along 79 half-scales.

WILKINSON (Wilkinson and Oates, 2016).[3] This dataset was also generated through crowdsourcing. Crowd workers were presented with small seed sets (e.g. *huge, small, microscopic*) and were asked to propose similar adjectives, resulting in twelve adjective sets. Sets were automatically cleaned for consistency, and then annotated for intensity by the crowd workers. The original dataset contains full scales. We use its division in 21 half-scales (with 61 adjective pairs) proposed by Cocos et al. (2018).

[1] http://demelo.org/gdm/intensity/
[2] https://github.com/acocos/scalar-adj
[3] https://github.com/Coral-Lab/scales

Dataset	Adjective scale
DEMELO	[*soft* < *quiet* < *inaudible* < *silent*]
	[*thick* < *dense* < *impenetrable*]
CROWD	[*fine* < *remarkable* < *spectacular*]
	[*scary* || *frightening* < *terrifying*]
WILKINSON	[*damp* < *moist* < *wet*]
	[*dumb* < *stupid* < *idiotic*]

Table 8.1: Examples of scales in each dataset. '||' denotes a tie between adjectives of the same intensity.

In the rest of this Chapter, we use the term "scale" to refer to the half-scales contained in these datasets. Table 8.1 shows examples from each one of them.

8.2.2 Sentence Collection

To explore the knowledge BERT has about relationships in an adjective scale s, we need to obtain a contextualised representation for every adjective $a \in s$. Since we are interested in comparing their intensity regardless of context, we want to avoid any effect coming from the specific contexts of use of each $a \in s$. We therefore generate a contextualised representation for each $a \in s$ in the same context. Since such cases are rare in running text, we construct two sentence sets that satisfy this condition using the ukWaC corpus (Baroni et al., 2009) and the Flickr 30K dataset (Young et al., 2014).[4] For every $s \in D$, a dataset from Section 8.2.1, and for each $a \in s$, we collect 1,000 instances (sentences) from each corpus.[5] We substitute each instance of an adjective a_i from scale s with $\forall\, a_j \in s$ where $j = 1...|s|$ and $j \neq i$, creating $|s| - 1$ new sentences.[6] For example, as illustrated in Figure 8.1, for an instance of *gorgeous* from the scale [*pretty* < *beautiful* < *gorgeous*] (e.g. "Punta Cana is *gorgeous*"), we generate two new sentences where *gorgeous* is replaced by each of the other adjectives (*pretty* and *beautiful*) in the same context ("Punta Cana is *pretty*" and "Punta Cana is *beautiful*").

The adjective substitution procedure just described may result in unnatural or incorrect sentences. We propose two ways to discard those:

Hearst patterns We filter out sentences with cases of specialisation or instantiation. For example, we want to avoid replacing *deceptive* with *fraudulent* and *false* in sentences like "*Viruses and other deceptive software*", "*Deceptive software such as viruses*", "*Deceptive software,*

[4]Flickr contains crowdsourced captions for 31,783 images describing everyday activities, events and scenes. We consider objective descriptions to be a better fit for our task than subjective statements, which might contain emphatic markers. For example, *impossible* would be a bad substitute for *impractical* in the sentence "*What you ask for is too impractical*".

[5]ukWaC has perfect coverage. Flickr 30K covers 96.56% of the DEMELO scales and 86.08% of the CROWD scales. A scale s is not covered when no $a \in s$ is found in a corpus.

[6]We make a minor adjustment of the substituted data by replacing the indefinite article *a* with *an* when the adjective that follows starts with a vowel, and the inverse when it starts with a consonant.

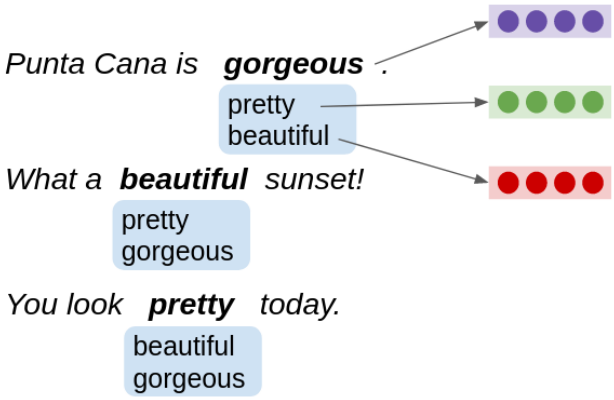


Figure 8.1: Illustration of the sentence collection procedure. We collect sentences containing an adjective a in a scale s (*pretty*, *beautiful*, *gorgeous*) from ukWaC and Flickr 30K and substitute a with all other adjectives in the scale.

especially viruses".[7] We parse the sentences with `stanza` (Qi et al., 2020) to reveal their dependency structure, and use Hearst lexico-syntactic patterns (Hearst, 1992) to identify sentences describing *is-a* relationships between nouns in a text. More details about this filtering are given in Appendix A.4.1.

Language Modelling criteria Adjectives that belong to the same scale might not be replaceable in all contexts. Polysemy can also influence their substitutability (e.g. *warm weather* is a bit *hot*, but a *warm smile* is *friendly*). In order to select contexts where $\forall a \in s$ fit, we measure the fluency of the sentences generated through substitution. We use a score assigned to each sentence by context2vec (Melamud et al., 2016) which reflects how well an $a \in s$ fits a context by measuring the cosine similarity between a and the context representation. We also experimented with calculating the perplexity assigned by BERT to a sentence generated through substitution, and with replacing the original a instance with the [MASK] token and getting the BERT probability for each $a \in s$ as a filler for that slot. context2vec was found to make better substitutability estimates. For this exploration, we use as development set a sample of 500 sentence pairs from the Concepts in Context (CoInCo) corpus (Kremer et al., 2014). Details on this evaluation and on the constitution of this sample are in Appendix A.4.2.

We use a 600-dimensional context2vec model in our experiments, pre-trained on ukWaC.[8] We calculate the context2vec score for all sentences generated for a scale s through substitution, and keep the ten sentences where the context2vec scores $\forall a \in s$ had the lowest standard deviation (STD). Low STD for a sentence means that $\forall a \in s$ are reasonable choices in this context. For comparison, we also randomly sample ten sentences from all the ukWaC sentences collected for each scale. We call the sets of sentences ukWaC, Flickr and Random SENT-SETS.

[7] Note that this would especially be a problem when considering adjectives with different polarity on a full scale (e.g. *deceptive* and *honest*).

[8] https://github.com/orenmel/context2vec

Method	Corpus	Sentences
Scale: *wrong → immoral → sinful → evil*		
context2vec-STD	**ukWaC**	I believe that war is *immoral*.
	Flickr	This boy was on the *wrong* end of this snowball fight.
Random	**ukWaC**	The author saw him and let him thru but not his mate as he had queued the *wrong* way.
Scale: *old → obsolete || outdated*		
context2vec-STD	**ukWaC**	(...) Chekhov was misunderstood and frequently seen by critics as merely an irreverent recorder of an *obsolete* way of life (...)
	Flickr	Two preschool aged boys are looking at an *old* locomotive.
Random	**ukWaC**	(...) rustic dialogue and good *old* fashioned laughter (...)

Table 8.2: Examples of sentences from our SENT-SETs selected with the context2vec-STD method compared to sentences randomly selected from ukWaC.

We extract the contextualised representation for each $a \in s$ in the ten sentences retained for scale s using the pre-trained `bert-base-uncased` model.[9] We do this for every BERT layer, which results in $|s| * 10 * 12$ BERT representations for each scale. Examples of the obtained sentences are given in Table 8.2.

8.2.3 Ranking with a Reference Point

In our first ranking experiment, we explore whether BERT encodes adjective intensity relative to a reference point, that is the adjective with the highest (or most extreme) intensity (a_{ext}) in a scale s. This is a pilot study to see if similarities derived from BERT representations encode some notion of intensity.

We rank $\forall a \in s$ where $a \neq a_{ext}$ by intensity by measuring the cosine similarity between their representation and that of a_{ext} in the ten ukWaC sentences retained for s, and in every BERT layer. For example, to rank [*thick, dense, impenetrable*] we measure the similarity of the representations of *thick* and *dense* to that of *impenetrable*. We then average the similarities obtained for each a and use these values for ranking (the more similar a is to a_{ext}, the more intense it is considered to be). We refer to this method as BERTSIM.

We evaluate the quality of the ranking for a scale by measuring its correlation with the gold standard ranking in the corresponding dataset D using Kendall's τ and Spearman's ρ correlation coefficients.[10] We also measure the model's pairwise accuracy (P-ACC) which shows whether it correctly predicted the relative intensity ($<$, $>$, $=$) for each pair a-$b \in s$ with $a \neq b$. During evaluation, we do not take into account scales where only one adjective is left ($|s| = 1$) after removing a_{ext} (26 out of 79 scales in CROWD; 9 out of 21 scales in WILKINSON; and none in DEMELO).

[9] When an adjective is split into multiple wordpieces (Wu et al., 2016), we average them to obtain its representation.

[10] As in Cocos et al. (2018), we report correlations as a weighted average using the number of adjective pairs in a scale as weights.

Dataset	Metric	BERTSIM	FREQ	SENSE
DEMELO	P-ACC	$\mathbf{0.591}_{11}$	0.571	0.493
	τ	$\mathbf{0.364}_{11}$	0.304	0.192
	ρ_{avg}	$\mathbf{0.389}_{11}$	0.309	0.211
CROWD	P-ACC	$\mathbf{0.646}_{11}$	0.608	0.570
	τ	$\mathbf{0.498}_{11}$	0.404	0.428
	ρ_{avg}	0.494_{11}	0.499	**0.537**
WILKINSON	P-ACC	$\mathbf{0.913}_{9}$	0.739_{9}	0.739_{9}
	τ	$\mathbf{0.826}_{9}$	0.478	0.586
	ρ_{avg}	$\mathbf{0.724}_{9}$	0.345	0.493

Table 8.3: BERTSIM results on each dataset using contextualised representations from the ukWaC SENT-SET. Subscripts denote the best-performing BERT layer.

We compare the BERTSIM method to two baselines which rank adjectives by frequency (FREQ) and number of senses (SENSE). We make the assumption that words with low intensity (e.g. *good, old*) are more frequent and polysemous than their extreme counterparts on the same scale (e.g. *awesome, ancient*). This assumption relies on the following two intuitions which we empirically validate:

(a) Extreme adjectives tend to restrict the denotation of a noun to a smaller class of referents than low intensity adjectives (Geurts, 2010). We hypothesise that extreme adjectives denote more exceptional and less frequently encountered properties of nouns than low intensity adjectives on the same scale (for instance, a *good view* is more common than a *fantastic view*). This is also reflected in the directionality of their entailment relationship (*fantastic* $\rightarrow$ *good*, *good* $\nrightarrow$ *fantastic*); low intensity adjectives should thus be more frequently encountered in texts. We test this assumption using frequency counts in Google Ngrams (Brants and Franz, 2006), and find that, in 75% of the scales, the least intense adjective is indeed more frequent than the most extreme adjective.

(b) Since frequent words tend to be more polysemous (Zipf, 1945), we also expect that low intensity adjectives would have more senses than extreme ones. This is confirmed by their number of senses in WordNet: in 67% of the scales, the least intense adjective has a higher number of senses than its extreme counterpart.

We present the results of this evaluation in Table 8.3. Overall, similarities derived from BERT representations encode well the notion of intensity, as shown by the moderate to high accuracy and correlation in the three datasets. The good results obtained by the FREQ and SENSE baselines (especially on CROWD) highlight the relevance of frequency and polysemy for scalar adjective ranking, and further validate our assumptions.

Figure 8.2 shows ranking predictions made by BERTSIM in different layers of the model. Predictions are generally stable and reasonable across layers, despite not always being correct. For example, the similarly-intense *happy* and *pleased* are inverted in some layers but are not

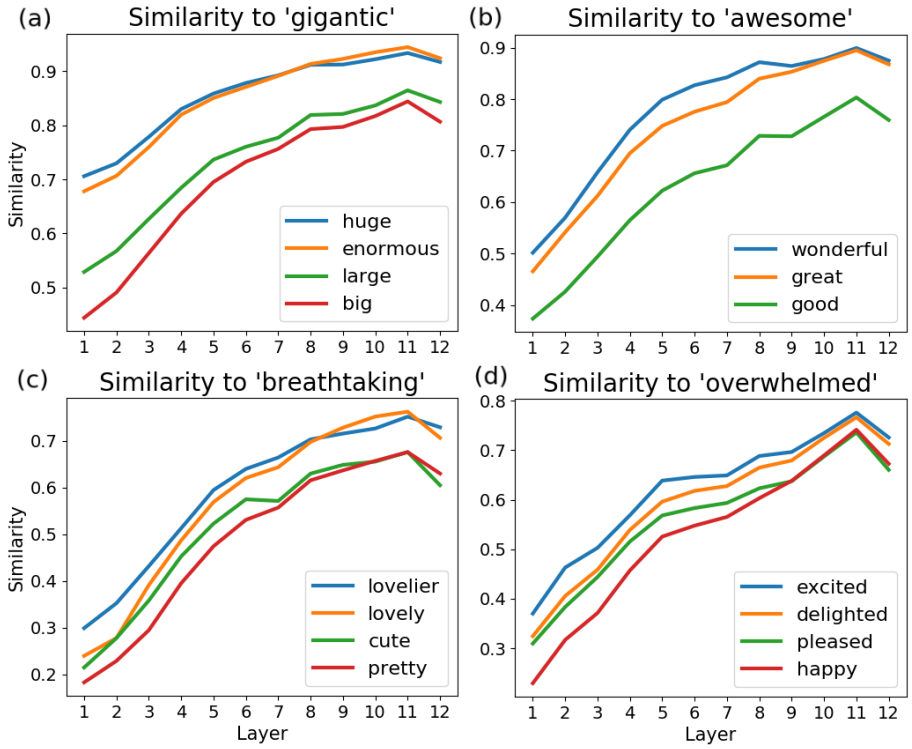


Figure 8.2: Examples of BERTSIM ranking predictions across layers using ukWaC sentences for four adjective scales: (a) [*big* < *large* < *huge* < *enormous* < *gigantic*], (b) [*good* < *great* < *wonderful* < *awesome*], (c) [*cute* < *pretty* < *lovely* < *lovelier* < *breathtaking*], (d) [*pleased* < *happy* < *excited* < *delighted* < *overwhelmed*]. (a) and (b) are from WILKINSON, (c) and (d) are from CROWD.

confused with adjectives further up the scale (*excited, delighted*). Note that *happy* and *pleased* are in adjacent positions in the CROWD ranking, and form a tie in the DEMELO dataset.

8.2.4 Ranking without Specified Boundaries: the DIFFVEC Method

In real life scenarios, scalar adjective interpretation is performed without concrete reference points (e.g. a_{ext}). We need to recognise that a *great book* is better than a *well-written* one, without necessarily detecting their relationship to *brilliant*.

Method Based on the encouraging results from the pilot experiment in the previous section, we developed a method that ranks adjectives based on their cosine similarity to a vector representing intensity. This method, called DIFFVEC, draws inspiration from word analogies in gender bias work, where a gender subspace is identified in word-embedding space by calculating the main direction spanned by the differences between vectors of gendered word pairs (e.g. $\overrightarrow{he}$ - $\overrightarrow{she}$, $\overrightarrow{man}$ - $\overrightarrow{woman}$) (Bolukbasi et al., 2016; Dev and Phillips, 2019; Ravfogel et al., 2020; Lauscher et al., 2020; Zhao et al., 2018).

We propose to obtain an **intensity vector** by subtracting the representation of a mild intensity adjective a_{mild} from that of an extreme adjective a_{ext} on the same scale. By subtracting *pretty* from *gorgeous*, for example, which express a similar core meaning (they are both on

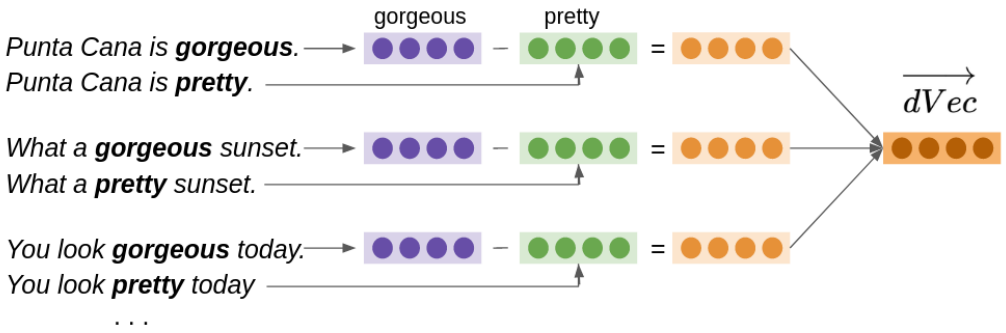


Figure 8.3: Simplified illustration of the procedure used for constructing $\overrightarrow{dVec}$ for one adjective pair from one scale using contextualised representations from a given layer.

the BEAUTY scale) but with different intensity, we expect the resulting $\overrightarrow{dVec} = \overrightarrow{gorgeous}$ - $\overrightarrow{pretty}$ embedding to represent this notion of intensity or degree. We can then compare other adjectives' representations to $\overrightarrow{dVec}$, and rank them according to their cosine similarity[11] to this intensity vector: the closer an adjective is to $\overrightarrow{dVec}$, the more intense it is.

We calculate the $\overrightarrow{dVec}$ for each $s \in D$ (a dataset from Section 8.2.1) using the most extreme (a_{ext}) and the mildest (a_{mild}) words in s. We experiment with BERT embeddings from the SENT-SETS generated through substitution as described in Section 8.2.2, and with static word2vec embeddings (Mikolov et al., 2013a) trained on Google News.[12] We build a $\overrightarrow{dVec}$ from every sentence (context) c in the set of ten sentences C for a scale s by subtracting the BERT representation of a_{mild} in c from that of a_{ext} in c. We average the ten $\overrightarrow{dVec}$'s obtained for s and construct a global $\overrightarrow{dVec}$ for the dataset D by averaging the vectors of $\forall s \in D$. For a fair evaluation, we do not build *and* evaluate $\overrightarrow{dVec}$ on the same dataset D. When evaluating on CROWD, we calculate a $\overrightarrow{dVec}$ vector on DEMELO (DIFFVEC-DM) and one on WILKINSON (DIFFVEC-WK), omitting all scales where a_{ext} or a_{mild} are present in CROWD. We do the same for the other datasets. Figure 8.3 illustrates the creation of $\overrightarrow{dVec}$ from one scale.

To obtain the $\overrightarrow{dVec}$ of a scale s with static embeddings, we simply calculate the difference between the word2vec embeddings of a_{ext} and a_{mild} in s.

Results For evaluation, we use the same metrics as in Section 8.2.3. We compare our results to the FREQ and SENSE baselines, and to the best results obtained by Cocos et al. (2018) who use information obtained from lexico-syntactic patterns, a lexicon annotated with intensity (SO-CAL) (Taboada et al., 2011), and paraphrases from PPDB.[13,14] Results are presented in Table 8.4. The DIFFVEC method gets remarkably high performance compared to previous results, especially when $\overrightarrow{dVec}$ is calculated with BERT embeddings. With the exception of

[11] We also tried the dot product of the vectors. The results were highly similar to the ones obtained using the cosine.

[12] We use the `magnitude` library Patel et al. (2018).

[13] We do not report Spearman's ρ from Cocos et al. (2018) because it was calculated differently: they measure it a single time for each dataset, treating each adjective as a single data point.

[14] In CROWD and WILKINSON, their best model combines the three types of information. The best-performing model on DEMELO relied only on information from patterns and a lexicon (P-ACC) or only from patterns (τ).

			DEMELO (DM)			CROWD (CD)			WILKINSON (WK)		
		Method	P-ACC	τ	ρ_{avg}	P-ACC	τ	ρ_{avg}	P-ACC	τ	ρ_{avg}
BERT	**ukWaC**	DIFFVEC-DM	-	-	-	$\mathbf{0.739}_{12}$	$\mathbf{0.674}_{12}$	$\mathbf{0.753}_{12}$	0.918_{6}	0.836_{6}	0.839_{6}
		DIFFVEC-CD	0.646_{8}	0.431_{8}	$\mathbf{0.509}_{8}$	-	-	-	0.869_{11}	0.738_{11}	0.829_{11}
		DIFFVEC-WK	0.584_{9}	0.303_{9}	0.313_{10}	0.706_{10}	0.603_{9}	0.687_{9}	-	-	-
	Flickr	DIFFVEC-DM	-	-	-	0.730_{12}	0.667_{12}	0.705_{10}	$\mathbf{0.934}_{9}$	$\mathbf{0.869}_{9}$	$\mathbf{0.871}_{9}$
		DIFFVEC-CD	0.620_{10}	0.377_{10}	0.466_{10}	-	-	-	0.902_{7}	0.803_{7}	0.798_{7}
		DIFFVEC-WK	0.579_{1}	0.294_{1}	0.321_{1}	0.702_{8}	0.608_{8}	0.677_{8}	-	-	-
	Random	DIFFVEC-DM	-	-	-	$\mathbf{0.739}_{12}$	0.673_{12}	0.743_{12}	0.918_{6}	0.836_{6}	0.839_{6}
		DIFFVEC-CD	0.626_{8}	0.388_{8}	0.466_{8}	-	-	-	0.836_{12}	0.672_{12}	0.790_{10}
		DIFFVEC-WK	0.557_{9}	0.246_{9}	0.284_{6}	0.703_{8}	0.598_{8}	0.676_{8}	-	-	-
word2vec		DIFFVEC-DM	-	-	-	0.657	0.493	0.543	0.787	0.574	0.663
		DIFFVEC-CD	0.633	0.398	0.444	-	-	-	0.803	0.607	0.637
		DIFFVEC-WK	0.593	0.323	0.413	0.618	0.413	0.457	-	-	-
Baseline		FREQ	0.575	0.271	0.283	0.606	0.386	0.452	0.754	0.508	0.517
		SENSE	0.493	0.163	0.165	0.658	0.498	0.595	0.721	0.586	0.575
		Cocos et al. '18	**0.653**	**0.633**	-	0.639	0.495	-	0.754	0.638	-

Table 8.4: Results of our DIFFVEC adjective ranking method on the DEMELO, CROWD, and WILKINSON datasets. We report results with contextualised (BERT) representations obtained from different SENT-SETs (ukWaC, Flickr, Random) and with static (word2vec) vectors. We compare to the frequency (FREQ) and number of senses (SENSE) baselines, and to results from previous work (Cocos et al., 2018). Results for a dataset are missing (-) when the dataset was used for building the $\overrightarrow{dVec}$ intensity vector.

Kendall's τ and pairwise accuracy on the DEMELO dataset, DIFFVEC outperforms results from previous work and the baselines across the board. We believe the lower correlation scores on the DEMELO dataset to be due to the large amount of ties present in this dataset: 44% of scales in DEMELO contain ties, versus 30% in CROWD and 0% in WILKINSON, where we obtain better results. Our models cannot easily predict ties using similarities which are continuous values. To check whether our assumption is correct, we make a simple adjustment to DIFFVEC so that it can propose ties if the vectors of two adjectives are similarly close to $\overrightarrow{dVec}$. Overall, this results in a small decrease in pairwise accuracy and a slight increase in correlation in DEMELO and CROWD. Complete results of this additional evaluation are given in Appendix A.4.3.

The composition of the SENT-SETs used for building BERT representations also plays a role on model performance. Overall, the selection method described in Section 8.2.2 offers a slight advantage over random selection, with ukWaC and Flickr sentences improving performance on different datasets. Note, however, that results for Flickr are calculated on the scales for which sentences were available (96.56% of DEMELO scales and 86.08% from CROWD).

The best-performing BERT layers are generally situated in the upper half of the Transformer network. The only exception is DIFFVEC-WK with the Flickr SENT-SET on DEMELO, where all layers perform similarly. The FREQ and SENSE baselines get lower performance than our method with BERT embeddings. SENSE manages to give results comparable to DIFFVEC with static embeddings and to previous work (Cocos et al., 2018) in one dataset (CROWD), but is still outperformed by DIFFVEC with contextualised representations.

We can also compare our results to those obtained by a purely pattern-based method on the

			DEMELO			CROWD		
		# Scales	P-ACC	τ	ρ_{avg}	P-ACC	τ	ρ_{avg}
BERT	**ukWaC**	1 (+)	0.653_{9}	0.438_{9}	0.489_{11}	$\mathbf{0.709}_{12}$	$\mathbf{0.611}_{12}$	0.670_{12}
		1 (−)	0.611_{10}	0.350_{10}	0.424_{11}	0.648_{10}	0.477_{10}	0.507_{10}
		5	0.650_{10}	0.430_{10}	0.514_{10}	0.700_{11}	0.595_{10}	0.673_{10}
	Flickr	1 (+)	0.656_{8}	0.449_{8}	0.504_{8}	0.676_{12}	0.552_{8}	0.612_{8}
		1 (−)	0.600_{3}	0.324_{3}	0.375_{5}	0.641_{9}	0.470_{9}	0.502_{9}
		5	0.647_{12}	0.426_{12}	0.498_{11}	0.692_{11}	0.587_{11}	0.640_{11}
	Random	1 (+)	$\mathbf{0.659}_{11}$	$\mathbf{0.451}_{11}$	0.493_{11}	0.691_{11}	0.570_{11}	0.658_{11}
		1 (−)	0.608_{12}	0.340_{12}	0.421_{10}	0.655_{10}	0.490_{10}	0.514_{12}
		5	0.653_{11}	0.442_{11}	$\mathbf{0.538}_{10}$	0.694_{11}	0.582_{11}	0.653_{11}
word2vec		1 (+)	0.602	0.334	0.364	0.624	0.419	0.479
		1 (−)	0.613	0.359	0.412	0.661	0.506	0.559
		5	0.641	0.415	0.438	0.688	0.559	0.601

Table 8.5: Results of DIFFVEC on DEMELO and on CROWD using a single positive (1 (+)) or negative (1 (−)) $a_{ext} - a_{mild}$ pair, and five pairs (5).

same datasets, reported by Cocos et al. (2018). This method performs well on DEMELO ($\tau = 0.633$) because of its high coverage on this dataset, which was compiled by finding adjective pairs that also match lexical patterns. The performance of the pattern-based method is much lower than that of our models in the other two datasets ($\tau = 0.203$ on CROWD, $\tau = 0.441$ on WILKINSON), and its coverage goes down to 11% on CROWD. This highlights the limitations of the pattern-based approach, as well as the efficiency of our model which combines high performance and coverage.

Further Exploration of DIFFVEC Given the high performance of the DIFFVEC method in the ranking task, we carry out additional experiments to explore the impact that the choice of scales and sentences has on the intensity vector quality. We test the method with a $\overrightarrow{dVec}$ vector built from a single $a_{ext} - a_{mild}$ pair of either positive (*awesome* − *good*) or negative (*horrible* − *bad*) polarity, that we respectively call DIFFVEC-1 (+)/(−). We also experiment with increasing the number of scales, adding *ancient-old*, *gorgeous-pretty* and *hideous-ugly* to form DIFFVEC-5. The scales are from WILKINSON, so we exclude this dataset from the evaluation.

Results are given in Table 8.5. We observe that a small number of word pairs is enough to build a $\overrightarrow{dVec}$ with competitive performance. Interestingly, DIFFVEC-1 (+) with random sentences obtains the best pairwise accuracy on DEMELO. The fact that the method performs so well with just a few pairs (instead of a whole dataset as in Table 8.4) is very encouraging, making our approach easily applicable to other datasets and languages.

A larger number of scales is beneficial for the method with static word2vec embeddings, which seem to better capture intensity on the negative scale. For BERT, instead, intensity modeled using a positive pair gives best results across the board. The use of five pairs of mixed

polarity improves results over a single negative pair, and has comparable performance to the single positive one.

Finally, we compare the performance of DIFFVEC-1 (+)/(−) and DIFFVEC-5 when the contextualised representations are extracted from a single sentence instead of ten. Our main observation is that reducing the number of sentences harms performance, especially when the sentence used is randomly selected. Detailed results are included in Appendix A.4.4.

8.2.5 Indirect Question Answering

We conduct an additional evaluation in order to assess how useful DIFFVEC adjective rankings can be in a real application. As in Cocos et al. (2018), we address Indirect Question Answering (QA) (de Marneffe et al., 2010). The task consists in interpreting indirect answers to YES/NO questions involving scalar adjectives. These do not straightforwardly convey a YES or NO answer, but the intended reply can be inferred. For example, if someone is asked "*Was it a good ad?*" and replies "*It was a great ad*", the answer is YES. This makes Indirect QA a good fit for scalar adjective ranking evaluation since it allows to directly assess a model's capability to detect the difference in intensity in an adjective pair.

We use the de Marneffe et al. (2010) dataset for evaluation, which consists of 125 QA pairs manually annotated with their implied answers (YES or NO). We adopt a decision procedure similar to the one proposed by de Marneffe et al. (2010). We compute the BERT embeddings of the adjective in the question (a_q) and the adjective in the answer (a_a). If a_a (e.g. *great*) has the same or higher intensity than a_q (e.g. *good*) the prediction is YES; otherwise, the prediction is NO. If the answer contains a negation, we switch YES to NO, and NO to YES. In previous work, indirect QA evaluation was performed on 123 or 125 examples, depending on whether cases labelled as "uncertain" were included (de Marneffe et al., 2010; Kim and de Marneffe, 2013; Cocos et al., 2018). de Marneffe et al. (2010)'s approach relies on a lexicon with intensity information automatically compiled from user reviews with associated ratings. Kim and de Marneffe (2013) use static embeddings (Mikolov et al., 2013b) and check whether the representation of a_a is closer to a_q or to an antonym of a_q retrieved from WordNet. We report available results from previous work, and our scores on the 123 YES/NO examples as in the most recent work by Cocos et al. (2018). We report results using DIFFVEC with an adjustment for ties, where two adjectives are considered to be of the same intensity if they are similarly close to $\overrightarrow{dVec}$ ($diffsim = \text{sim}(\overrightarrow{dVec}, \overrightarrow{a_q}) - \text{sim}(\overrightarrow{dVec}, \overrightarrow{a_a})$). If $|diffsim|$ (the absolute value of $diffsim$) < 0.01, we count them as a tie. We also compare our method to FREQ and SENSE, and to a baseline predicting always the majority label (YES). Results of this evaluation are given in Table 8.6. The best performance is obtained when $\overrightarrow{dVec}$ is obtained from the Wilkinson dataset (DIFFVEC-WK). DIFFVEC with BERT embeddings consistently outperforms the baselines and de Marneffe et al. (2010)'s approach, and presents a clear advantage over DIFFVEC with static word2vec representations. Several configurations surpass also Cocos et al. (2018)'s method, but only DIFFVEC-WK achieves higher performance than the model of Kim and de Marneffe (2013).

		Method	Acc	P	R	F
BERT	ukWaC	DIFFVEC-1 $(+)_{10}$	0.715	0.677	0.692	0.685
		DIFFVEC-DM$_{12}$	0.707	0.670	0.689	0.678
		DIFFVEC-CD$_{12}$	0.675	0.635	0.648	0.642
		DIFFVEC-WK$_{11}$	**0.740**	**0.712**	**0.739**	**0.725**
	Flickr	DIFFVEC-1 $(+)_{9}$	0.699	0.663	0.680	0.672
		DIFFVEC-DM$_{11}$	0.699	0.659	0.673	0.666
		DIFFVEC-CD$_{10}$	0.691	0.653	0.667	0.660
		DIFFVEC-WK$_{5}$	0.683	0.646	0.661	0.654
	Random	DIFFVEC-1 $(+)_{9}$	0.715	0.677	0.692	0.685
		DIFFVEC-DM$_{10}$	0.724	0.691	0.713	0.702
		DIFFVEC-CD$_{12}$	0.667	0.629	0.642	0.636
		DIFFVEC-WK$_{11}$	0.699	0.667	0.688	0.677
word2vec		DIFFVEC-1 (+)	0.667	0.633	0.650	0.641
		DIFFVEC-DM	0.602	0.554	0.559	0.557
		DIFFVEC-CD	0.593	0.548	0.553	0.551
		DIFFVEC-WK	0.585	0.543	0.547	0.545
Baselines		FREQ	0.593	0.548	0.553	0.551
		SENSE	0.593	0.560	0.568	0.564
		MAJ	0.691	0.346	0.500	0.409
Previous		de Marneffe et al. (2010)	0.610	0.597	0.594	0.596
		Kim and de Marneffe (2013)	0.728	0.698	0.714	0.706
		Cocos et al. (2018)	0.642	0.710	0.683	0.684

Table 8.6: Results of our DIFFVEC method with contextualised (BERT) and static (word2vec) embeddings on the indirect QA task. We compare to the frequency, polysemy and majority baselines, and to results from previous work.

8.2.6 Discussion

Our initial exploration of the knowledge encoded in BERT representations about scalar adjectives using the BERTSIM method (Section 8.2.3) showed they can successfully serve to rank them by intensity. Our DIFFVEC method (Section 8.2.4) outperformed BERTSIM, providing even better ranking predictions with as few resources as a single adjective pair. This difference can be explained by the nature of the vectors used in the two settings. The a_{ext} representation in BERTSIM contains information about the meaning of the extreme adjective alongside its intensity, while the $\overrightarrow{dVec}$ vector is a cleaner representation of intensity. The subtraction of $\overrightarrow{a_{mild}}$ from $\overrightarrow{a_{ext}}$ removes information about the core meaning expressed by their scale (e.g. BEAUTY, TEMPERATURE, SIZE). The DIFFVEC method can estimate adjectives' relative intensity on the fly without using any external knowledge source, a requirement needed in previous approaches. Notably, one of its highest performing variants (DIFFVEC-1 (+)) makes high quality predictions with a vector constructed from a single adjective pair.

We hypothesised that the sentences used for extracting BERT representations would need

to be natural contexts for all adjectives in a scale. This, however, has not been confirmed by our evaluation. Precisely, differences between our methods when relying on carefully vs randomly selected sentences are minor. This might be due to several reasons. Although BERT representations are contextualised, they also encode knowledge about the meaning and intensity of words acquired through pre-training, independent of the new contexts of use. Another possible explanation is that due to the skewed distribution of word senses (Kilgarriff, 2004; McCarthy et al., 2004), a high proportion of our randomly selected sentences probably contain instances of the adjectives in their most frequent sense. If this is also the meaning of the corresponding scale, then there are high chances that the sentences be a good fit. Finally, it is also possible that the quality of our carefully selected sentences is not high enough to provide a clear advantage over randomly chosen ones, especially when using multiple sentences per scale.

The DIFFVEC-1 (+) method with BERT embeddings, which uses a vector derived from a single positive pair, yields consistently better results than DIFFVEC-1 (−) which relies on a single negative pair. To better understand this difference in performance, we examine the composition of the DEMELO and CROWD datasets, specifically whether there is an imbalance in terms of polarity as reflected in the frequency of positive vs negative adjectives. We check the polarity of the adjectives in two sentiment lexicons: SO-CAL (Taboada et al., 2011) and AFINN-165 (Nielsen, 2011). The two lexicons cover a portion of the adjectives in DEMELO and CROWD: 68% and 79%, respectively. The DEMELO dataset is well-balanced in terms of positive and negative adjectives: 51% and 49% of the covered adjectives fall in each category. In CROWD, we observe a slight skew towards positive: 61% vs 39%. According to this analysis, the difference in performance between the two methods cannot be fully explained by an imbalance in terms of polarity.

We perform an additional analysis based on the Google Ngram frequency of the positive and negative words that were used for deriving DIFFVEC. The adjectives *good* (276M) and *awesome* (10M) are more frequent than *bad* (65M) and *horrible* (4M), respectively. In fact, we find that the 1,000 most frequent positive words in SO-CAL and AFINN are, on average, much more frequent (18M) than the 1,000 most frequent negative words (8M). Word frequency has a direct impact on word representations, since having access to sparse information about a word's usages does not allow the model to acquire rich information about its linguistic properties as in the case of frequent words (Luong et al., 2013; Schick and Schütze, 2020). The high frequency of *good* and *awesome* results in better quality representations than the ones obtained for their antonyms, and could explain to some extent the improved performance of DIFFVEC-1 (+) compared to DIFFVEC-1 (−) with BERT embeddings. However, this analysis does not explain the difference in the performance of DIFFVEC (+) and (−) between BERT and word2vec. This would require a better understanding of how words with different polarity (antonyms) are represented in BERT's space compared to word2vec. We leave these explorations for future work.

Regarding the performance of different BERT layers, we observe that knowledge relevant for scalar adjective ranking is situated in the last layers of the Transformer network. Figure 8.4

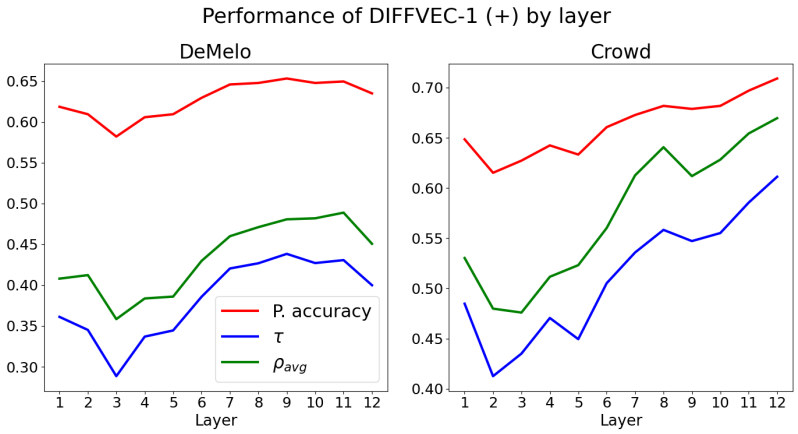


Figure 8.4: Performance of DIFFVEC-1 (+) with ukWaC sentences across BERT layers.

shows how the performance of DIFFVEC-1 (+) changes across different BERT layers: model predictions improve after layer 3, and performance peaks in one of the last four layers. This contrasts with Vulić et al. (2020)'s observation that type-level lexical knowledge is predominantly located in earlier layers. In that study, the contexts used to derive word representations are different for every word. This context variation probably has a larger impact on the upper layer representations, where contextualisation is stronger (Ethayarajh, 2019). The different contexts used, however, are not relevant for the word type-level tasks they evaluate on (e.g. word similarity and lexical relation prediction, among others), which may explain why earlier layers perform better in their study. We instead compare representations of words occurring in the same contexts, ruling out context variation, and find that upper layers perform better in this setting. Another possible explanation for the different layer behaviour in the two studies would be the differences between the aspects of lexical meaning being addressed.

The DIFFVEC method is simple, effective and requires very few resources, which makes it easy to apply it to other languages. Reliance on external resources and evaluation datasets for scalar adjective ranking has, however, restricted research to English. In the next section, we explain how we compiled a new multilingual dataset to extend the method to other languages and to promote further research on them.

8.3 Scalar Adjective Ranking in Other Languages

In this section, we present MULTI-SCALE, a new dataset for the evaluation of scalar adjective ranking methods in French, Spanish and Greek. We investigate whether intensity information is also encoded in monolingual and multilingual BERT representations in these languages, and set performance baselines on the dataset.

	DEMELO
EN	dim < gloomy < dark < black
FR	terne < sombre < foncé < noir
ES	sombrío < tenebroso < oscuro < negro
EL	αμυδρός || αχνός < μουντός < σκοτεινός< μαύρος
	WILKINSON
EN	bad < awful < terrible < horrible
FR	mauvais < affreux < terrible < horrible
ES	malo < terrible < horrible < horroroso
EL	κακός < απαίσιος < τρομερός < φρικτός

Table 8.7: Example translations from each dataset. "||" indicates adjectives at the same intensity level (ties).

8.3.1 The MULTI-SCALE Dataset

To build the MULTI-SCALE dataset, we translate the DEMELO (de Melo and Bansal, 2013) and WILKINSON (Wilkinson and Oates, 2016) datasets, which contain 87 and 21 half-scales, respectively. Adjective scales were manually translated to French, Spanish and Greek by two speakers with native or near-native proficiency of each language. They were shown the adjectives in the context of a scale. This context narrows down the possible translations for polysemous adjectives to the ones that express the meaning described inside the scale. For example, the Spanish translations proposed for the adjective *hot* in the scales [*warm* < *hot*] and [*flavorful* < *zesty* < *hot* || *spicy*] are *caliente* and *picante*, respectively. Additionally, the translators were instructed to preserve the number of words in the original scales when possible. In some cases, however, they proposed multiple translations for English words, or none if an adequate translation could not be found. As a result, the translated datasets have a different number of words and ties. In a few cases, translators proposed prepositional or adverbial phrases (FR: *en surpoids* "with excess weight" for *overweight*; ES: *mal parecido* "bad-looking" for *unattractive*). We include these in our experiments as well. Table 8.7 shows examples of scales in each language and Table 8.8 contains statistics on the composition of the translated datasets.

We have seen that a random selection of sentences works well enough for this task (cf. Section 8.2.4), in spite of adjectives in a scale not always being interchangeable. We collect French, Spanish and Greek sentences containing the adjectives from OSCAR (Suárez et al., 2019), a corpus derived from CommonCrawl. French, Spanish and Greek are morphologically rich languages where adjectives need to agree with the noun they modify. To keep the method resource-light, we gather sentences that contain the adjectives in their unmarked form. For each scale s, we randomly select ten sentences from OSCAR where adjectives from s occur. Then, we generate additional sentences through lexical substitution as in Section 8.2.2: for every sentence (context) c that contains an adjective a_i from scale s, we replace a_i with $\forall\ a_j \in s$ where $j = 1...|s|$ and $j \neq i$. This process results in a total of $|s|$ * 10 sentences per scale.

		# unordered pairs	# adjectives
DEMELO	EN	548 (524)	339 (293)
	FR	590 (567)	350 (303)
	ES	448 (431)	313 (275)
	EL	557 (535)	342 (295)
WILKINSON	EN	61 (61)	59 (58)
	FR	67 (67)	61 (60)
	ES	59 (59)	58 (56)
	EL	68 (68)	61 (58)

Table 8.8: Content of the translated datasets, with the number of unique adjectives and pairs in parentheses.

For English, we use the ukWaC-Random set of sentences (Section 8.2.2).

8.3.2 Methodology

We apply the DIFFVEC method (Section 8.2.4) to the MULTI-SCALE dataset. We build an intensity representation using a single positive adjective pair (DIFFVEC-1 (+)) in each language, which has given highly competitive results in English. The pairs we use are the translations of a_{mild} and a_{ext} in a positive scale (*perfect-good*) from the CROWD dataset.[15] We also learn a $\overrightarrow{dVec}$ representation by averaging the $\overrightarrow{dVec}$s of all (a_{mild}, a_{ext}) pairs in WILKINSON that do not appear in DEMELO (DIFFVEC-WK), and another one from pairs in DEMELO that are not in WILKINSON (DIFFVEC-DM).

Models We conduct experiments with state-of-the-art contextual models and several baselines on the MULTI-SCALE dataset. We use the pre-trained multilingual BERT model (Devlin et al., 2019) and report results of the best model (between `cased` and `uncased`) for each language. We also report results obtained with the following monolingual models:

- `bert-base-uncased` (Devlin et al., 2019) for English;
- `flaubert_base_uncased` (Le et al., 2020) for French;
- `bert-base-spanish-wwm-uncased` for Spanish (Cañete et al., 2020);
- `bert-base-greek-uncased-v1` (Koutsikakis et al., 2020) for Greek.

We feed the collected sentences to each model and extract the representations corresponding to all a's in a scale s from every layer of the model. When an adjective is split into multiple wordpieces, we average the representations of all pieces (we call this approach "WP") or all pieces but the last one ("WP-1"). The intuition behind this is that the last part of a word often corresponds to a suffix that carries morphological information.

[15] FR: parfait-bon, ES: perfecto-bueno, EL: τέλειος-καλός.

We compare the contextual models to monolingual fastText static embeddings in each language (Grave et al., 2018).[16] We also compare our results to the frequency and polysemy baselines (FREQ and SENSE) described in Section 8.2.3. We have seen these are strong baselines for English, and we expect the intuitions behind them (i.e. that words with mild intensity are more frequent and more polysemous than words with extreme intensity) to hold across languages. For French, Spanish and Greek, frequency is taken from OSCAR. The number of senses is retrieved from BabelNet (Navigli and Ponzetto, 2012) for Spanish and French.[17] For adjectives that are not present in BabelNet, we use a default value which corresponds to the average number of senses for adjectives in the dataset (DEMELO or WILKINSON) for which this information is available. We omit the SENSE baseline for Greek due to low coverage. Only 46.7% of Greek adjectives have a BabelNet entry, compared to 95.7% and 88.9% of Spanish and French adjectives in our datasets.

8.3.3 Results

We use the same evaluation metrics as in Sections 8.2.3 and 8.2.4: pairwise accuracy (P-ACC), Kendall's τ and Spearman's ρ. Results on this task are given in Table 8.9.

Monolingual models perform consistently better than the multilingual model, except for French. We report the best wordpiece approach (WP or WP-1) for each model: WP-1 works better with all monolingual models and the multilingual model for English. Using all wordpieces (WP) is a better choice for the multilingual model in other languages. We believe that WP-1 is not better in these cases because the multilingual wordpiece vocabulary is mostly English-driven, resulting in highly arbitrary partitionings in these languages (e.g. ES: *fantástico* → fant-ástico; EL: γιγάντιος (*gigantic*)→γ-ι-γ-άν-τιος). Tokenisers of the monolingual models instead tend to split words in a way that more closely reflects the morphology of the language (e.g. ES: *fantástico* → fantás-tico; EL: γιγάντιος→γιγά-ντι-ος. Detailed results of this comparison are found in Appendix A.4.5.

We observe that DIFFVEC-1 (+) yields comparable and sometimes better results than DIFFVEC-DM and DIFFVEC-WK, which are built from multiple pairs. This is important especially in the multilingual setting, since it shows that just one pair of adjectives in a new language is enough for obtaining good results. The best layer varies across models and configurations. The monolingual French and Greek models generally obtain best results in earlier layers, and so does the multilingual model for English to some extent, whereas the other models improve in the upper half (layers 6-12). This shows that the semantic information relevant for adjective ranking is not situated at the same level of the Transformer in different languages. The lower results in French can be due to the higher amount of ties in the datasets compared to other languages.[18] The baselines obtain competitive results, confirming that the underlying linguistic intuitions hold across languages. The best models beat the baselines in

[16] https://fasttext.cc/docs/en/crawl-vectors.html

[17] We omit Named Entities from BabelNet entries – for example, names of TV shows or locations– because their meaning is often very specific and not widely known.

[18] 58% of the French DEMELO scales contain a tie, compared to 45% in English.

		EN			FR			ES			EL		
		Mono WP-1			**Mono WP-1**			**Mono WP-1**			**Mono WP-1**		
		P-ACC	τ	ρ_{avg}	P-ACC	τ	ρ_{avg}	P-ACC	τ	ρ_{avg}	P-ACC	τ	ρ_{avg}
DM	DV-1 (+)	**.651**$_9$	**.435**$_9$	**.496**$_9$	**.610**$_3$	**.369**$_3$	**.396**$_3$	.658$_9$	.381$_9$	**.407**$_9$	.564$_2$	.238$_1$	.271$_2$
	DV-WK	.586$_6$	.267$_6$	.300$_6$	.515$_1$	.167$_1$	.166$_7$	**.670**$_7$	**.404**$_7$	**.407**$_7$	.589$_2$	.294$_2$	.325$_2$
WK	DV-1 (+)	.852$_1$	.705$_1$	.802$_1$	.612$_6$	.257$_6$	.215$_6$	**.814**$_7$	**.627**$_7$	**.803**$_9$	.618$_8$	.282$_8$	.256$_8$
	DV-DM	**.918**$_{10}$	**.836**$_{10}$	**.859**$_{10}$	.642$_7$	.322$_2$	.392$_2$	.780$_6$	.559$_6$	.684$_6$	**.750**$_{10}$	**.564**$_{10}$	**.586**$_{10}$
		Multi WP-1			**Multi WP**			**Multi WP**			**Multi (unc) WP**		
DM	DV-1 (+)	.609$_4$	.346$_4$	.389$_4$	.559$_7$	.260$_7$	.311$_7$	.614$_3$	.291$_3$	.268$_5$	.517$_9$	.139$_9$	.163$_9$
	DV-WK	.544$_3$	.208$_3$	.241$_4$	.517$_{10}$	.170$_{10}$	.179$_{10}$	.618$_{12}$	.301$_{12}$	.303$_{12}$	.539$_9$	.181$_9$	.207$_9$
WK	DV-1 (+)	.836$_6$	.672$_6$	.717$_6$	.672$_3$	.382$_3$	.380$_3$	.797$_3$	.593$_3$	.639$_3$	.662$_{10}$	.388$_9$	.423$_9$
	DV-DM	.836$_7$	.672$_7$	.766$_7$	**.701**$_6$	**.441**$_6$	**.476**$_2$	.695$_{10}$	.390$_{10}$	.511$_{10}$	.691$_5$	.447$_5$	.502$_5$
		Static models and baselines											
DM	DV-1 (+)	.637	.407	.458	.573	.288	.275	.656	.383	**.421**	.575	.266	.273
	DV-WK	.599	.330	.406	.454	.033	-.006	.616	.298	.315	.549	.205	.217
	FREQ	.575	.271	.283	.602	.346	.345	.585	.227	.239	**.596**	**.306**	**.334**
	SENSE	.493	.163	.165	.512	.229	.185	.516	.139	.151	-	-	-
WK	DV-1 (+)	.787	.574	.663	.582	.197	.152	.695	.390	.603	.706	.464	.566
	DV-DM	.852	.705	.783	.642	.325	.280	.712	.424	.547	.691	.447	.451
	FREQ	.754	.508	.517	.567	.167	.148	.576	.153	.382	.676	.417	.427
	SENSE	.721	.586	.575	.567	.255	.340	.644	.411	.456	-	-	-

Table 8.9: Results of the DIFFVEC (DV) method with monolingual (Mono) and multilingual (Multi) contextual models. Comparison to static embeddings and baselines per language. Subscripts denote the best layer. The best result obtained for each dataset in each language is indicated in boldface. For all languages but Greek, the multilingual model is cased.

all configurations except for Greek on the DEMELO dataset, where FREQ and static embeddings obtain higher results. Overall results are lower than those reported for English, which shows there is room for improvement in new languages.

8.4 Scalar Adjective Identification

8.4.1 The SCAL-REL dataset

Previous work focused on scalar adjective ranking in pre-compiled resources, and this has also been the case in our experiments. However, the decision of whether an adjective expresses intensity or not is a crucial one. In settings where an intensity-based analysis can be beneficial (such as QA and recommendation systems), it is important to identify adjectives where the notion of intensity applies, and distinguish them from relational adjectives. We hereby propose a dataset for this new task in English because of the possibility to automatically compile a dataset in this language.

SCAL-REL contains relational adjectives, labelled as "pertainyms" in WordNet, and scalar adjectives from the DEMELO, WILKINSON and CROWD datasets. We include all unique scalar adjectives in the datasets (443 in total) and keep the same number from the 4,316 unique such adjectives labelled with the pertainym relationship in WordNet (Fellbaum, 1998), including

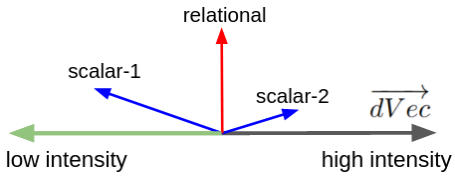


Figure 8.5: Illustration of two scalar adjectives that are close to $\overrightarrow{dVec}$ and to its opposite (which represents low intensity). The red vector describes a relational adjective that is perpendicular to $\overrightarrow{dVec}$.

many rare or highly technical terms (e.g. *birefringent*, *anaphylactic*).[19] Scalar adjectives in our datasets are much more frequent than these relational adjectives; their average frequency in Google Ngrams is 27M and 1.6M, respectively. We balance the relational adjectives set by frequency, by subsampling 222 frequent and 221 rare adjectives. We use the mean frequency of the 4,316 relational adjectives in Google Ngrams as a threshold.[20] We propose a train/dev/test split of the dataset (65/10/25%), ensuring that the two classes are balanced in each subset. To obtain contextualised representations, we extract ten random sentences from ukWaC for each pertainym; for scalar adjectives, we use the ukWaC-Random sentence pool (cf. Section 8.2.2).

8.4.2 Methodology

For each English adjective in the SCAL-REL dataset, we generate a representation from the available ten sentences (cf. Section 8.4.1) using the `bert-base-uncased` model. We use the two wordpiece approaches described in Section 8.3.2 (WP and WP-1). We experiment with a simple logistic regression classifier that uses the averaged representation for an adjective (ADJ-REP) as input and predicts whether it is scalar or relational. We also apply the DIFFVEC-1 (+) method to this task and measure how intense an adjective is by calculating its cosine with $\overrightarrow{dVec}$. The absolute value of the cosine indicates how clearly an adjective encodes the notion of intensity. In Figure 8.5, we show two scalar adjective vectors with negative and positive cosine similarity to $\overrightarrow{dVec}$, and another vector that is perpendicular to $\overrightarrow{dVec}$, i.e. describing a relational adjective for which the notion of intensity does not apply.[21] We train a logistic regression model to find a cosine threshold separating scalar from relational adjectives (DV-1 (+)). Finally, we also use as a feature the cosine similarity of the adjective representation to the vector of "*good*", which we consider as a prototypical scalar adjective (PROTO-SIM). The best BERT layer is selected based on the accuracy obtained on the development set. We report accuracy on the test set. The baseline classifiers only use frequency (FREQ) and polysemy (SENSE) as features. We use these baselines on SCAL-REL because the WordNet pertainyms included in the dataset are rarer than the scalar adjectives. The intuition behind the SENSE

[19]Note that the WordNet annotation does not cover all pertainyms in English (for example, frequent words such as *ironic* or *seasonal* are not marked with this relation).

[20]Nine scalar adjectives from our datasets are also annotated as pertainyms in WordNet (e.g. *skinny, microscopic*) because they are denominal. We consider these adjectives to be scalar for our purposes since they clearly belong to intensity scales.

[21]To draw a parallel with gender debiasing, this value would reveal words' bias in the gender direction Bolukbasi et al. (2016), regardless of the gender (male or female).

Method	Accuracy	
	WP	WP-1
ADJ-REP (BERT)	**0.946**$_9$	0.942$_9$
PROTO-SIM	0.888$_{11}$	0.902$_{10}$
DV-1 (+)	0.549$_2$	0.545$_2$
ADJ-REP (fastText)	0.929	
FREQ	0.669	
SENSE	0.714	

Table 8.10: Classification results on the SCAL-REL dataset.

baseline explained in Section 8.2.3 also applies here.

8.4.3 Evaluation

Results on this task are given in Table 8.10. The classifier that relies on ADJ-REP BERT representations can distinguish the two types of adjectives with very high accuracy (0.946), closely followed by fastText embeddings (0.929). The DV-1 (+) method does not perform as well as the classifier based on ADJ-REP, which is not surprising since it relies on a single feature (the absolute value of the cosine between $\overrightarrow{dVec}$ and ADJ-REP). Comparing ADJ-REP to a typical scalar word (PROTO-SIM) yields better results than DV-1 (+). The SENSE and FREQ baselines can capture the distinction to some extent. Relational adjectives in our training set are less frequent and have fewer senses on average (2.59) than scalar adjectives (5.30). A closer look at the errors of the best model reveals that these concern tricky cases: one of the four misclassified scalar adjectives is derived from a noun (*microscopic*), whilst five out of eight wrongly classified relational adjectives can have a scalar interpretation (e.g. *sympathetic, imperative*). Overall, supervised models obtain very good results on this task. SCAL-REL will enable research on unsupervised methods that could be used in other languages.

8.5 Conclusion

We have shown that BERT representations encode rich information about the intensity of scalar adjectives which can be efficiently used for their ranking. Our proposed method, DIFFVEC, is simple and resource-light, solely relying on an intensity vector which can be derived from as few as a single example. In spite of its simplicity, it outperforms previous work on the scalar adjective ranking and Indirect Question Answering tasks. Our performance analysis across BERT layers highlights that the lexical semantic knowledge needed for these tasks is mostly located in the higher layers of the BERT model.

We created a new scalar adjective dataset for French, Spanish and Greek and applied our methodology to these languages, experimenting with monolingual BERT models and mBERT. Our results show that BERT representations encode rich information about the semantics of

scalar adjectives in different languages. Additionally, we propose a new classification task and a benchmark dataset that can serve to estimate the models' capability to distinguish between scalar and relational adjectives. A supervised BERT-based model does very well on this task, and can thus be used to identify lexical items that contribute to the emotional load and meaning of a text.

The experiments presented in this chapter open up new avenues for future research on intensity, polarity, and other connotational aspects of lexical meaning. It would be interesting to explore adjective ranking in full scales (instead of half-scales) and evaluate the capability of contextualised representations to detect polarity, antonyms, and even different emotions (e.g. sadness or anger) (Mohammad, 2018). It would also be worth investigating how negation affects BERT representations, and to perform intensity-based ranking in scales containing negated adjectives (*not gorgeous* $\overset{?}{\rightarrow}$ *pretty*) (Gotzner et al., 2018).

Another question that remains open is how to choose good candidates for building $\overrightarrow{dVec}$. Our experiments do not show a clear trend in this respect. The best performing pairs might vary depending on the evaluation dataset, the type of embeddings used and the connotations of the respective adjectives in each language. We need to carry out experiments involving many more adjective pairs in order to detect conclusive patterns. That could be, for example, that more frequent or positive adjectives are better alternatives for creating $\overrightarrow{dVec}$.

In this work we have focused on adjectives, but we can find similar intensity relations in other parts of speech: verbs (*adore* > *love*), nouns (*downpour* > *rain*) and adverbs (*furiously* > *angrily*). Another possible extension of this work would involve investigating whether our adjective-based $\overrightarrow{dVec}$ can be useful for ranking words of other parts of speech by intensity, or for building a specific $\overrightarrow{dVec}$ for each part of speech.

Finally, the DIFFVEC method can potentially be applied to other dimensions of difference between near-synonyms, such as formality and complexity. Just as we obtain a representation of intensity from two adjectives that have the same meaning but differ in intensity, one could obtain a formality representation from, for example, the subtraction of the vectors of *father* and *dad* ($\overrightarrow{father} - \overrightarrow{dad}$). Words could then be ranked according to their formality or complexity by reference to this representation.

Chapter 9

Nouns' Semantic Properties and their Prototypicality

9.1 Introduction

In the previous chapter, we investigated the knowledge encoded in BERT about semantic relationships, specifically addressing the intensity relationship between scalar adjectives. In this chapter, we address another aspect of adjectival meaning, namely their role as modifiers in adjective-noun (AN) constructions. We probe BERT for noun properties and their prototypicality, as expressed by the adjectives that modify them in AN phrases. This study is focused on English because of the availability of datasets that can be used for evaluation.

Adjectival modification is one of the main types of composition in natural language (Baroni and Zamparelli, 2010; Guevara, 2010). Adjectives in attributive position[1] usually have a restrictive role on the reference scope of the noun they modify, limiting the set of things it refers to (e.g. *white rabbits* $\sqsubset$ *rabbits*). This property of adjectives has interesting entailment implications, generally leading to AN constructions where the entailment relationship with the head noun holds (AN $\models$ N) (Baroni et al., 2012). The entailment relationship is unidirectional (*white rabbit* $\models$ *rabbit* but *rabbit* $\not\models$ *white rabbit*) (Kotlerman et al., 2010), unless modification is not restrictive: when A is prototypical of the N it modifies (as in *soft silk, red strawberry*), its insertion does not reduce the scope of N or add new information, but rather emphasises some inherent property of N (Pavlick and Callison-Burch, 2016). In these cases, N and AN denote the same set and are in an equivalence relation (*red strawberry* = *strawberry*). Entailment between these pairs is symmetric, in contrast to the restrictive case.

Alongside the theoretical interest of this linguistic property and its impact on the entailment properties of AN constructions, identifying prototypical adjectives can be useful in practical applications. It can serve to retrieve information about the general concept (*silk, strawberry*) when queries include such ANs (*soft silk, red strawberry*) or the other way around, to retrieve

[1] Adjectives that appear immediately before the noun they modify and form part of the noun phrase (*white rabbit*), as opposed to adjectives in predicative position that occur after the noun (this *rabbit* is *white*).

information from sources containing the AN when the query contains N. It can also serve to discard adjectives that do not add new information about the noun they modify in summarisation or sentence compression.

We investigate the knowledge BERT encodes about nouns' inherent properties as described in AN constructions. We use a set of collected norms that describe important concept features (McRae et al., 2005) and their associated quantifiers (Herbelot and Vecchi, 2015). We rely on these data to derive cloze statements that we use to query BERT about noun properties, and to train BERT-based classifiers predicting these properties. We furthermore fine-tune BERT for entailment and test it in a task that involves AN constructions (Pavlick and Callison-Burch, 2016). For this experiment, we rely on the AddOne dataset proposed by Pavlick and Callison-Burch (2016) which consists of sentence pairs that contain AN pairs annotated for entailment in a crowdsourcing task. The proposed simplified entailment task only differs from the classical recognising textual entailment (RTE) task (Dagan et al., 2005) in that the premise (p) and hypothesis (h) differ by one atomic edit e (i.e. insertion of A).

Compositionality in AN constructions has been a central topic in distributional and formal semantics. Mitchell and Lapata (2010) derive the meaning representation of a composite phrase from that of its constituents by performing algebraic operations (addition and multiplication) on distributional word semantic vectors, while Baroni and Zamparelli (2010) and Guevara (2010) derive composite vectors through composition functions learned from corpus-harvested phrase vectors. Baroni et al. (2012) also demonstrate that the entailment relationship that exists between AN phrases and their head N (*big cat* ⊨ *cat*) transfers to lexical entailment among nouns (*dog* ⊨ *animal*). In our work, we represent AN phrases by combining the contextualised BERT representations of A and N in sentences where they occur using algebraic operations. We also investigate the extent to which the representations of A and N in an AN capture its meaning, since token-level BERT embeddings encode information from the surrounding context.

Prototypicality has been addressed in the literature mainly by reference to relationships between nouns, i.e. the typical hyponyms in a specific semantic class (e.g. *dog* ⊨ *animal*) or member concepts that are most central to a category (Roller and Erk, 2016). Vulić et al. (2017) also address verb prototypicality in terms of how typical of an action a verb is (e.g. "Is TO RUN a type of TO MOVE?"). The prototypicality of adjectives with respect to nouns has been understudied and is absent from lexico-semantic resources such as WordNet (Fellbaum, 1998) and HyperLex (Vulić et al., 2017).

On the probing side, previous work explores the factual and common sense knowledge present in pretrained language models (LMs) using "fill-in-the-blank" cloze statements (Petroni et al., 2019; Jiang et al., 2020). The `HasProperty` relation in the LAMA benchmark (Petroni et al., 2019) (cf. Chapter 2.3.1), extracted from ConceptNet (Speer and Havasi, 2012), is similar to our relation of interest as it links nouns to adjectives describing their properties. ConceptNet contains 3,894 such pairs, but a close inspection of the data reveals several problematic cases (e.g. *informal both*, *divine forgive*, *ten 10*). Additionally, the cloze statements proposed for this

dataset were automatically extracted from Open Mind Common Sense (OMCS)[2] sentences and are often very long, including irrelevant information.[3] Other studies probing BERT with cloze statements (Jiang et al., 2020; Bouraoui et al., 2020; Ettinger, 2020; Ravichander et al., 2020) do not explore noun properties.

Our results show that BERT has limited knowledge of noun properties and their prevalence, but can still successfully detect cases where the addition of an adjective does not alter the meaning of a sentence and where entailment is preserved.

9.2 Datasets

McRae et al. (2005) dataset (MRD) Semantic feature norms are used in the field of psycholinguistics for studying human semantic representation and computation. MRD contains feature norms for 541 living and nonliving concepts collected from 725 participants in an annotation task. The annotators proposed features they thought were important for each concept, covering physical (perceptual), functional and other properties. Among the collected 7,258 concept-feature pairs, we find that a dolphin is *intelligent, friendly*, and *lives in oceans*, and that a chandelier is *hanging from ceilings* and *is made of crystal*. The number of annotators who proposed each feature is also provided. The dataset has been extensively used to investigate and improve the knowledge about object properties encoded by distributional models (Rubinstein et al., 2015), static word embeddings (Lucy and Gauthier, 2017; Yang et al., 2018) and, more recently, contextualised LMs (Forbes et al., 2019; Hasegawa et al., 2020). These studies do not focus on adjectival attributes but rather consider all proposed properties, or specific subsets such as visual properties. In our experiments, we explore noun properties through the "IS_ADJ" features of noun concepts in MRD.

Herbelot and Vecchi (2015) dataset (HVD) HVD adds an extra level of quantification annotations to the MRD norms. Three native speakers of English select a natural language quantifier among [NO, FEW, SOME, MOST, ALL][4] for each concept-feature (C, f) pair, expressing the ratio of C instances having feature f (e.g. ALL guitars are musical instruments, but SOME guitars are electric). Subject-predicate quantification is important for semantic inference; it can serve to understand set relations (e.g. synonymy and hyponymy) and to derive logically entailed sentences for a statement. We use the HVD dataset in our study to probe BERT about the prevalence of noun properties.

Pavlick and Callison-Burch (2016) dataset (AddOne) The Addone dataset is focused on AN composition. It contains 5,560 sentence pairs involving an AN pair (s_N, s_{AN}) which have

[2] https://github.com/commonsense/omcs

[3] For example: "To understand the event "The monkey ate some bananas.", it is important to know that Banana is [MASK]". The ground truth adjective in this case is *yellow*.

[4] NO and FEW labels were rarely used by the annotators and we consider them as describing cases of non typical attributes.

been manually annotated for entailment ($s_N \models s_{AN}$) by crowd workers. Addone sentence pairs differ by one atomic edit, the insertion of A:

- s_N: "There are questions as to whether our culture has changed."
- s_{AN}: "There are questions as to whether our traditional culture has changed."

Sentences were collected from corpora of different genres and each pair is annotated with a score in a 5-point scale from 1 (contradiction) to 5 (entailment). Only the pairs with high agreement (same score assigned by 2 out of 3 annotators) were retained. We use the AddOne dataset to assess BERT's ability to detect entailment in AN constructions. The dataset comes with a pre-defined split into training, development and test sets (83/10/7%) which we use in our experiments addressing entailment (Section 9.5).

9.3 Cloze Task Experiments

We probe BERT for noun properties (Section 9.3.1) and their prototypicality (Section 9.3.2) with cloze statements. We use the `bert-base-uncased` and `bert-large-uncased` models pre-trained on the BookCorpus (Zhu et al., 2015) and on English Wikipedia (Devlin et al., 2019).

9.3.1 Cloze Task Probing for Properties

We retrieve adjective modifiers of nouns in MRD found in the IS_ADJ features describing a concept (*bouquet*: IS_*colourful*; *panther*: IS_*black*). There are 509 noun concepts with at least one IS_ADJ feature in MRD. We exclude features involving multi-word attributes (*coconut*: IS_*white_inside*, *raft*: IS_*tied_together_with_rope*) which we do not expect BERT to be able to predict. The average number of features per noun is 3.12 (1,592 in total). Table 9.1 shows the number of nouns having a specific number of features. We define a set of templates and derive cloze statements for each noun that serve as our queries to probe BERT for these attributes. We define templates using both its singular and plural forms, as shown in Table 9.2.[5] We always use plural templates for nouns given in plural form in MRD,[6] and singular templates for mass and uncountable nouns.[7] We evaluate the quality of the predictions made by BERT for each slot by checking the presence of ground-truth (gold) MRD adjectives at positions @1, @5 and @10, i.e. the top one, five and ten predictions ranked by probability. We compare BERT to a baseline that ranks by frequency all bigrams where a specific noun appears in the second position ("___ *bouquet*") in Google Ngrams (Brants and Franz, 2006), excluding bigrams that contain stop words[8] and punctuation.

[5] We use the plural form of nouns given by the `pattern` Python library and manually correct any errors.

[6] The following 26 nouns: beans, beets, curtains, earmuffs, jeans, leotards, mittens, onions, pajamas, peas, scissors, skis, slippers, shelves, sandals, bolts, gloves, nylons, boots, screws, pants, tongs, trousers, drapes, pliers, socks.

[7] There are three such nouns in MRD: rice, bread, football.

[8] We use NLTK's (Bird et al., 2009) list of English stop words.

# attributes	1	2	3	4	5	6	7	8	9
# nouns	98	124	97	76	60	35	12	6	1

Table 9.1: Number of nouns with a specific number of IS_ADJ attributes in MRD. In total, there are 509 nouns with 1,592 attributes.

Masking Properties	
singular	a balloon is [MASK].
plural	balloons are [MASK].
usually	a balloon is usually [MASK]. balloons are usually [MASK].
generally	a balloon is generally [MASK]. balloons are generally [MASK].
can be	a balloon can be [MASK]. balloons can be [MASK].
most	most balloons are [MASK].
all	all balloons are [MASK].
some	some balloons are [MASK].
Masking Quantifiers	
[MASK] balloons are colourful.	(ALL-MOST-SOME)
[MASK] balloons are large.	(SOME-SOME-FEW)
[MASK] balloons are round.	(MOST-SOME-NO)

Table 9.2: Cloze statements for the noun *balloon* with its properties (McRae et al., 2005) and quantifiers masked. Parentheses in the lower part of the table contain the quantifiers proposed by annotators in the (Herbelot and Vecchi, 2015) dataset.

The two plots in Figure 9.1 show the number of nouns for which the BERT-base (top plot) and BERT-large (lower plot) models manage to propose at least one correct attribute from MRD at the first, top five or top ten positions in the ranking. We observe that results differ considerably when different cloze statements from Table 9.2 are used for probing. Overall, templates that contain the noun in singular form (`N [usually|generally] is [MASK], N can be [MASK]`) cause BERT to suggest correct attributes for far less nouns than templates containing the noun in plural form (`Ns [usually|generally] are [MASK],Ns can be [MASK]; [most|all|some] Ns are [MASK]`). Notably, the frequency baseline proposes more correct properties than BERT-base when probed with templates that contain the noun in singular form. The highest number of nouns that receive at least one correct attribute is 287 (out of 509) and is obtained with most-P queries (`most Ns are [MASK]`) and BERT-large. For BERT-base, usually-P queries (`Ns usually are [MASK]`) retrieve at least one correct attribute in @10 for 222 nouns. Correct attributes are more rarely found at higher positions, with only a small number of nouns being assigned one at the first position (@1). In Appendix A.5.1 we additionally report recall values of this experiment, calculated over the words for which at least one correct attribute is found.

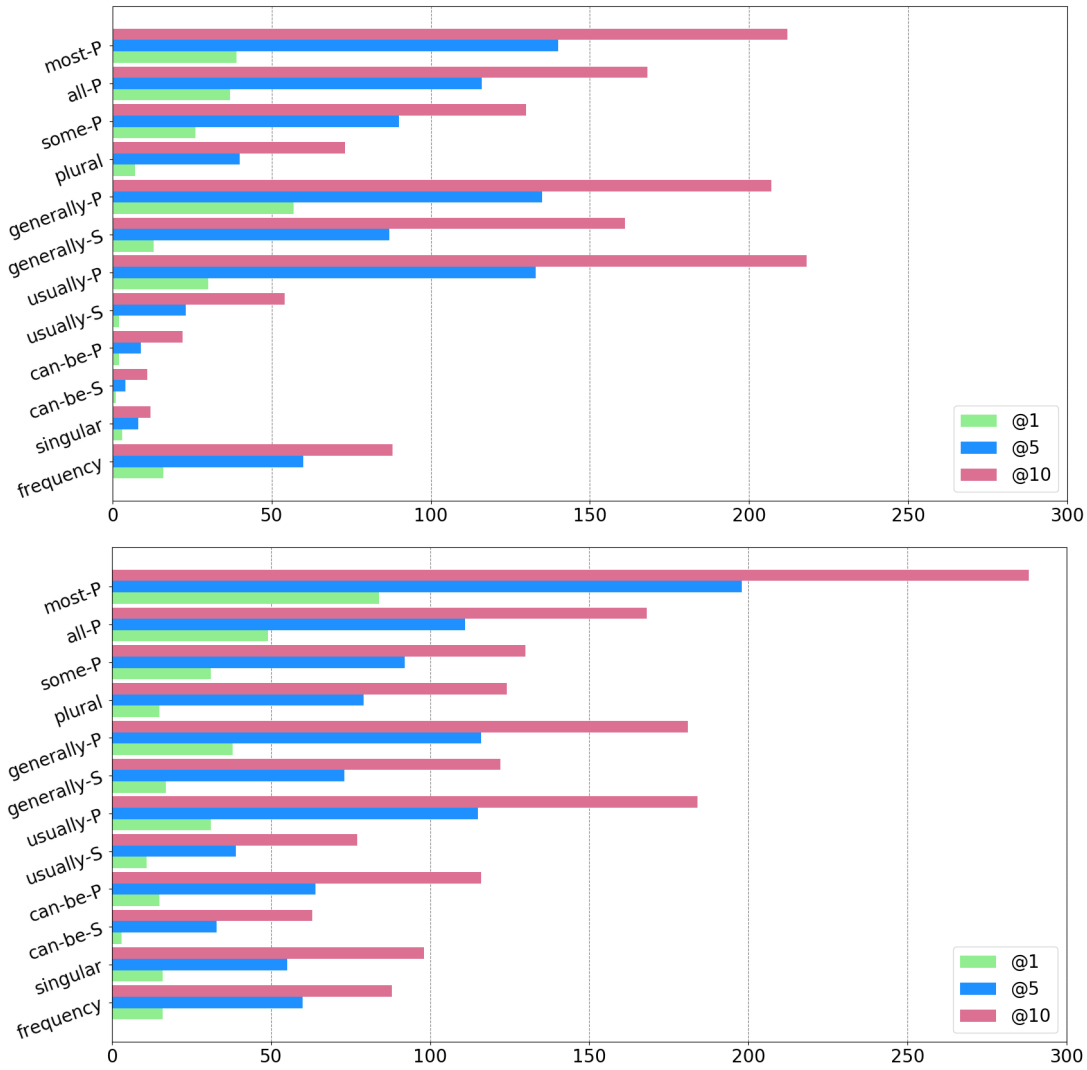


Figure 9.1: Number of nouns (out of 509) for which a correct (gold) attribute is found at positions @1, @5 and @10 of the ranked BERT predictions, when using sentences constructed with the templates on the y axis. S and P denote templates with the noun in singular or plural form (cf. Table 9.2). The top figure shows results for BERT-`base` and the lower one for the BERT-`large` model.

These results suggest that BERT has marginal knowledge of noun properties as reflected in the MRD association norms. This cloze task is more challenging than others targeting encyclopedic knowledge (Petroni et al., 2019), probably because information about noun properties is not as often explicitly stated in text. We however observe that the quality of the proposed adjectives is quite high in some cases, even when these are not present in MRD and cannot thus be captured by this evaluation. For example, the predictions retrieved with the probe "mittens are generally [MASK]" describe different aspects of the noun such as their colour (*white, black, red, yellow*), shape and composition (*flat, thick, short, thin*), and the fact that they can be *removed*. This shows that BERT encodes some knowledge about the noun being a garment, although it fails to guess the specific adjectives chosen by the annotators in MRD (*knitted* and *colourful*). Naturally, prediction quality varies a lot and in some cases these

do not describe noun properties but general knowledge about the described entity, as seen in the predictions obtained with the probe "all balloons are [MASK]": *empty, free, flown, filled, lit, inflated, green, destroyed, closed, used*.

9.3.2 Cloze Task Probing for Quantifiers

We additionally probe BERT's ability to predict how general a noun property is, i.e. whether it is prototypical and affects all or most members of the class of objects referred to by the noun, or a subset of it. We create cloze statements from HVD where the quantifier is masked but the property is present (e.g. [MASK] bananas are healthy.). Since this task explores the set of objects to which a property applies, we form the cloze statements using the plural form of the noun. We query BERT using these statements and check whether it correctly predicts the missing quantifier.

We evaluate BERT's predictions against the annotations in HVD. We split the data into Set (A) which contains AN pairs that have at least two ALL, or a combination of ALL and MOST, annotations; and Set (B) which contains all other pairs that were assigned SOME, FEW and NO labels. We view AN pairs in (A) as prototypical, characterising the entire class (e.g. *banana*-IS_*healthy* → [ALL-ALL-ALL]), and AN pairs in (B) as describing properties that apply to a subset of the objects described by the noun (e.g, *apple*-IS_*red* → [MOST-SOME-SOME]). We create 788 cloze statements of the form "`[MASK] Plural_Noun are A`" for 386 nouns in Set (A), and 808 statements for 391 nouns in Set (B).[9] We retrieve the first ten BERT suggestions for filling the masked slot in the cloze statements, and evaluate their quality by checking whether the quantifiers are among the predictions (precision at 10). When all quantifiers are proposed, we additionally check their relative position in the ranking, i.e. if ALL and MOST precede SOME in (A) predictions, and if SOME comes first in (B) predictions.

The results are shown in Table 9.3. We observe that all three quantifiers are frequently in the top ten predictions for most statements in both sets, which suggests that BERT is not capable of distinguishing prototypical from other noun properties, at least with this probing task targeting quantifier prediction. If BERT encoded knowledge about the prevalence of properties for nouns in the queries, we would have expected to find ALL and MOST more often than SOME in the results for Set (A), and SOME more often than ALL and MOST in the @10 predictions for Set (B). The precedence of a quantifier over another, shown in the lower part of the table, leads to the same conclusion. In order to infer that BERT encodes prototypicality information, ALL and MOST should be higher ranked than SOME in Set (A) predictions and the inverse in Set (B), but this does not seem to be the case.

[9]Note that a noun might be present in both Sets (A) and (B), depending on whether the ANs where it is involved describe prototypical properties. We find, for example, "*jar* IS_*transparent*" in Set A, because all jars have this property, and "*jar* IS_*breakable*" in Set B, because not all jars can be easily broken.

BERT-base			
	Set A		**Set B**
ALL @10	627/788	SOME @10	571/808
MOST @10	508/788	ALL @10	608/808
SOME @10	623/788	MOST @10	445/808
ALL < SOME	298/532	SOME < ALL	161/467
MOST < SOME	225/451	SOME < MOST	26/31
BERT-large			
	Set A		**Set B**
ALL @10	592/788	SOME @10	528/808
MOST @10	494/788	ALL @10	612/808
SOME @10	548/788	MOST @10	477/808
ALL < SOME	255/462	SOME < ALL	150/449
MOST < SOME	250/431	SOME < MOST	12/25

Table 9.3: Frequency of appearance of a quantifier in the top ten ranked BERT-`base` and BERT-`large` model predictions for the 788 sentences in Set (A) and the 884 sentences in Set (B). "<" denotes precedence of a quantifier over another, when they both appear in @10. For example, ALL precedes SOME in the ranking for 298 Set (A) predictions out of 532 where they have both been proposed by BERT-`base`.

9.4 Classification Experiments

We probe BERT representations for prototypicality also in a classification setting, using frozen embeddings and fine-tuning. In these experiments, we use only the `bert-base-uncased` model.

9.4.1 Experimental Setup

Examples We consider as positive (prototypical) instances (`pos`) for this task AN phrases from HVD Set (A), with at least two ALL or a combination of MOST and ALL annotations (cf. Section 9.3.2). As negative instances (`neg`) for a noun in (A), we use the AN pairs where it appears in Set (B). If |`neg`| < |`pos`| for an N, we collect additional negative instances from the ukWaC corpus (Baroni et al., 2009) where N is modified by an adjective A′ such that A′N ∉ HVD. We exclude cases where N is part of a compound (i.e. where it modifies another noun, as in *small sardine tin*).[10] We retain the most frequent ANs found for N in ukWaC as negative instances, until |`neg`| = |`pos`|. The dataset contains 1,566 instances in total, 783 for each class.[11]

[10]We obtain the dependency parse of a sentence using `stanza` Qi et al. (2020)

[11]We omit five positive AN pairs because not enough negative instances were found for the noun in Set (B) or in ukWaC.

Representations For each AN in |pos| and |neg|, we obtain a BERT representation from a sentence (s_{AN}) in ukWaC where A modifies N. We pair s_{AN} with a sentence s_N where A has been automatically deleted (e.g. s_{AN}: "Then shape into balls about the size of a small tangerine" vs. s_N: "Then shape into balls about the size of a tangerine"). We choose sentences where A is not modified by an adverb (e.g. *very small ant*, where removing *small* would result in an ungrammatical sentence). When no sentences are found for an AN (588 out of 1,566 cases), we use as s_{AN} the plural pattern from the cloze task experiments (e.g. *raspberries are edible*; cf. Section 9.3.1) and the plural noun alone as s_N (*raspberries*). When N is an uncountable noun, we use the singular pattern instead.[12] More details about how BERT representations are extracted from these sentences for each type of experiment are found in Sections 9.4.2 and 9.4.3.

Data split We keep aside 10% of the data as our development set and perform 5-fold cross-validation on the rest. To minimise the impact of lexical memorisation where the model learns that a word is representative of a specific class (Levy et al., 2015), we observe a full lexical split by adjective between the development set and the data used for cross-validation, and also between the training and the test set in each fold. As a result, adjectives found in the test set at each iteration have not been seen in the training or in the development set. This is done to avoid that the model memorises an adjective as describing a common or prototypical property of nouns (e.g. *small* is a feature for 120 out of 509 nouns in MRD). The split allows to evaluate the capability of the model to generalise to unseen adjectives.

9.4.2 Embedding-based Classification

We expect the vector of an AN phrase involving a prototypical adjective (*red strawberry*) to be more similar to the vector of N (*strawberry*), than that of a phrase A′N involving an adjective that expresses a non typical property of N (*rotten strawberry*). We extract three types of BERT embeddings from each layer of the `bert-base-uncased` model that we use to compare the representation of an AN to that of the head N:

1. an embedding for N in sentence s_N (where N occurs without the adjective) ($\overrightarrow{Ns_N}$);
2. an embedding for N in sentence s_{AN} (which contains the adjective) ($\overrightarrow{Ns_{AN}}$);
3. an embedding for A in s_{AN}.

We obtain an AN representation by combining the vectors pairwise: $\overrightarrow{Ns_N}$ and $\overrightarrow{Ns_{AN}}$; $\overrightarrow{Ns_N}$ and $\overrightarrow{As_{AN}}$; $\overrightarrow{Ns_{AN}}$ and $\overrightarrow{As_{AN}}$, using different composition operations: average, concatenation, difference, multiplication, and addition. We also experiment with the token-level contextualised representations $\overrightarrow{As_{AN}}$ and $\overrightarrow{Ns_{AN}}$ in isolation which we expect to also encode information about the noun and the adjective in the AN, respectively, since they occur in the same context. We use the different AN representations as features for a logistic regression classifier.

[12] Using sentences created with these patterns for all ANs hurts performance compared to the setting where sentences gathered from corpora are used.

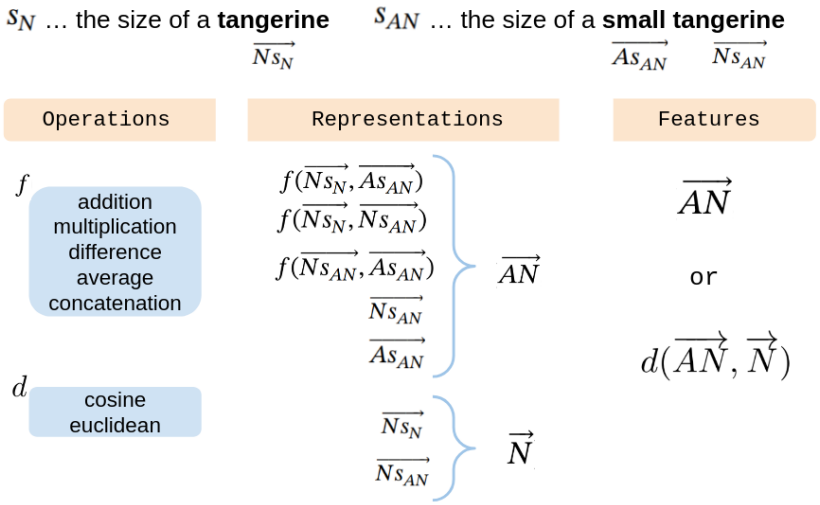


Figure 9.2: Illustration of the types of features used to train the classifiers. We create $\overrightarrow{AN}$ representations from different vectors using different operations (f). The classifier uses the resulting representations as features, or the distance/similarity (d) between $\overrightarrow{AN}$ and a representation of the noun ($\overrightarrow{N}$).

Additionally, we calculate the cosine similarity and euclidean distance between the representation of a noun ($\overrightarrow{Ns_N}$ or $\overrightarrow{Ns_{AN}}$) and $\overrightarrow{AN}$ obtained through the vector combinations and composition operations described above, and feed them to the classifier as individual features or in combination. Figure 9.2 contains an illustration of the different features we use to train the classifiers. For comparison, we also run experiments using static word2vec (Mikolov et al., 2013a) and fastText (Grave et al., 2018) embeddings as features, creating $\overrightarrow{AN}$ with the word embeddings $\overrightarrow{N}$ and $\overrightarrow{A}$, and using $\overrightarrow{A}$ alone. For each type of representation (BERT, word2vec, fastText), we select the configuration with the highest average accuracy on the development set over the five cross-validation runs.

In Table 9.4, we report the average accuracy (Acc) and F1 score on the test sets of the five folds for these configurations. Accuracy is calculated over all examples in the test set. Precision (P), recall (R) and F1-score show how good a model is at detecting AN pairs that involve a prototypical adjective. As baselines, we provide results for a model that always predicts prototypicality (ALL-PROTO), and a model that assigns the majority label found in the training set at each fold (MAJORITY).

In terms of accuracy, BERT obtains the best results on this task (0.658) when cosine similarity and euclidean distance between $\overrightarrow{Ns_N}$ and $\overrightarrow{Ns_N}$+$\overrightarrow{Ns_{AN}}$ at the last (12th) layer are used as features. The simple ALL-PROTO baseline obtains the highest F1 score (0.672) but a low accuracy in this balanced dataset. Static representations, especially word2vec, perform worse than BERT but still manage to beat the baselines in terms of accuracy. The best configuration for word2vec and fastText was the use of the static adjective representations ($\overrightarrow{A}$) as features, which shows that the models do not manage to extract the information needed for assessing prototypicality from the different $\overrightarrow{N}$ and $\overrightarrow{A}$ combinations. Instead, the best strategy is to learn the tendency of an adjective to be prototypical. When evaluated on unseen adjectives in our

Model	Acc	F1	P	R
BERT	**0.658**	0.648	**0.676**	0.633
fastText	0.593	0.481	0.639	0.411
word2vec	0.559	0.455	0.601	0.372
ALL-PROTO	0.507	**0.672**	0.507	**1.000**
MAJORITY	0.473	0.524	0.390	0.800

Table 9.4: Average accuracy (Acc), F1-score, precision (P) and recall (R) of embedding-based classifiers on the HVD dataset in the cross-validation experiment across five folds.

$\overrightarrow{AN}$ type	Acc	composition	Acc
Ns_N, Ns_{AN}	0.712	addition	0.712
As_{AN}	0.675	difference	0.667
Ns_N, As_{AN}	0.667	concatenation	0.660
Ns_{AN}, As_{AN}	0.665	average	0.650
Ns_{AN}	0.613	multiplication	0.611

Table 9.5: Highest average accuracy obtained by the different types of AN representation (left) and composition operations (right) with BERT embedding-based classifiers on the HVD development set.

test sets, they base prototypicality judgments on the similarity of these adjectives to the ones seen in the training set. We observe a high variation in accuracy and F1 scores across folds for all models. For BERT, F1 scores range from 0.553 to 0.740 and the range is even larger for the fastText-based model (from 0.310 to 0.747). This suggests that prototypicality is not easy to detect for all AN pairs. Overall, BERT contextualised embeddings seem to be a better fit for estimating prototypicality than static representations.

We explore the behaviour of different kinds of features on the development set. In Table 9.5, we report the best results obtained for each type of BERT-based $\overrightarrow{AN}$ representation and composition operation. The combination of Ns_N and Ns_{AN} clearly outperforms the other vector combinations. Using the adjective token-level representation alone ($\overrightarrow{As_{AN}}$) also yields good results, definitely higher than $\overrightarrow{Ns_{AN}}$. In terms of composition functions, addition is the best performing operation for this task and multiplication the least useful. We report the detailed results by layer, and the best configurations per $\overrightarrow{AN}$ and composition type in Appendix A.5.2.

9.4.3 Fine-tuning BERT

We compare our results in the frozen- embedding experiments with performance of BERT fine-tuned for the prototypicality task. Specifically, we feed into BERT the two sentences in each (s_N, s_{AN}) pair separated by the [SEP] token. We experiment with a classifier on top of the [CLS] token, as is typically done in sentence-pair classification tasks with BERT (we call this approach BERT-CLS); and with a classifier on top of the concatenation of two token representations: ($\overrightarrow{Ns_N}$, $\overrightarrow{As_{AN}}$), ($\overrightarrow{Ns_N}$, $\overrightarrow{Ns_{AN}}$), ($\overrightarrow{Ns_N}$, $\overrightarrow{As_{AN}} + \overrightarrow{Ns_{AN}}$) (our BERT-TOK approach). The two classification heads consist of a linear layer with softmax and are trained with a cross

Model	Acc	F1	P	R
BERT-CLS	**0.700**	**0.654**	0.772	**0.579**
BERT-TOK	0.696	0.642	**0.777**	0.551

Table 9.6: Average accuracy, F1 score, precision and recall in the cross-validation experiment across five folds for a BERT model fine-tuned on the HVD dataset using the CLS and TOK approaches.

entropy loss. We fine-tune each model for 3 epochs with 0.1 dropout, and choose the learning rate based on the accuracy on the development set. Results of these experiments are found in Table 9.6. BERT-CLS and BERT-TOK ($\overrightarrow{Ns_N}$, $\overrightarrow{As_{AN}}$) perform comparably on this task and obtain better results than embedding-based models (Table 9.4), with 0.697 accuracy.

9.5 Entailment in AN Constructions

9.5.1 Task Description

AN constructions are often in a forward entailment relation with the head noun (*white rabbit* ⊨ *rabbit*) (Baroni et al., 2012).[13] Whether backward entailment holds depends on the properties of N described by A in AN. For example, a *car* is not always *red* (the label would be "Unknown"), while *strawberry* always entails *red strawberry*. We explore BERT's capability to identify the AN cases where backward (N ⊨ AN) entailment holds[14] using the Addone dataset (Pavlick and Callison-Burch, 2016) (cf. Section 9.2).

We fine-tune BERT on Addone to assess whether it captures the entailment relationship involved in AN constructions. BERT has shown high performance in other textual entailment tasks Devlin et al. (2019), but the Addone dataset has proved challenging for other models relying on RNN and LSTM architectures. We follow Pavlick and Callison-Burch (2016) and use Addone for a binary classification task, with the labels ENTAILMENT (for forward entailment and equivalence) and NOT ENTAILMENT (encompassing the contradiction, independence and reverse entailment relations). Similarly to the fine-tuning approach described in Section 9.4.3, we feed into BERT the two sentences in each pair (s_N, s_{AN}) separated by the special [SEP] token. We again use the CLS and TOK classification heads. We fine-tune the model for 5 epochs with 0.1 dropout and select the learning rate based on the F1 score calculated over the actual ENTAILMENT cases on the development set.[15]

[13] An exception to this are ANs with non-subsective adjectives, such as *former* (*former president* ⊭ *president*.)

[14] Backward entailment (N ⊨ AN) holds when A denotes a prototypical property of N, and also when A emphasises that the whole of N is involved (e.g. *chicken* ⊨ *whole chicken*)

[15] We use F1 score as a criterion, and not accuracy, because the Addone dataset is highly imbalanced (only 23% of instances belong to the ENTAILMENT class).

Model	Acc	F1	P	R
Human (P&CB)	0.933	0.730	0.840	0.640
MAJ-BY-ADJ (P&CB)	**0.922**	0.680	**0.860**	0.560
MAJ (P&CB)	0.853	-	-	-
BERT-TOK	0.912	**0.696**	0.709	0.684
BERT-CLS	0.147	0.257	0.147	**1.000**
RNN (P&CB)	0.873	0.510	0.600	0.440

Table 9.7: Results on the Addone test set. We highlight in boldface the best results obtained by the models and the baselines. We include results and baselines reported by Pavlick and Callison-Burch (2016) (P&CB) for comparison. Human performance determines the upper bound that can be obtained for this task.

9.5.2 Results

Results of our experiments on Addone are presented in Table 9.7. We include results reported by Pavlick and Callison-Burch (2016) for comparison. We report the accuracy, F1 score, precision and recall obtained by each model. The MAJ and MAJ-BY-ADJ baselines assign the majority class in the training set (NON-ENTAILMENT) and the majority class proposed for each adjective in the training set, respectively. We also report the human performance on this task as an upper bound, and compare to the best-performing model in Pavlick and Callison-Burch (2016) which relies on a RNN architecture (Bowman et al., 2015). BERT-CLS fails to learn the information needed for the task and predicts the ENTAILMENT label for all instances. The default fine-tuning strategy used for textual entailment with BERT is, thus, not suitable for addressing cases of compositional entailment in the Addone dataset. It is much more effective to use the representations of the specific words that determine sentence entailment: BERT-TOK ($\overrightarrow{Ns_N}$, $\overrightarrow{As_{AN}}$) obtains higher results than the previous best model (RNN) and beats the MAJ baseline, as well as MAJ-BY-ADJ in terms of F1 and recall.

9.6 Conclusion

We have proposed a thorough investigation of the information encoded by BERT about nouns' intrinsic properties as expressed by adjectives in AN constructions. This topic has only marginally been explored in previous work, mainly in the frame of studies addressing the model's relational and encyclopedic knowledge. Using datasets specifically compiled for psycholinguistics studies, we have probed BERT for noun properties and their prototypicality, and have explored the entailment relationship that holds between nouns and the AN construction where they can appear. Our cloze task experiment results show that BERT encodes limited knowledge about noun properties and their prevalence, as described in word association norms. It is important to note that these results are tied to the specific properties proposed by annotators in the McRae dataset. Different annotation procedures (for example, a cloze task) might lead to a different set of attributes. In a supervised setting, however, BERT can learn to distinguish

prototypical from other noun properties. When fine-tuned on data specifically addressing the N ⊨ AN entailment relationship, BERT manages to beat previous best performing models and strong baselines on this task.

Chapter 10

Conclusion

10.1 Contributions

At the beginning of this thesis, we set out two main goals: investigating the lexical semantic knowledge encoded in context-sensitive representations derived from neural language models, and improving the quality of the information encoded in the representations. The main contributions and findings with respect to each of these goals are outlined below.

Main Findings In order to fulfill the first goal, we performed extensive experiments exploring different aspects of word meaning. The investigated aspects can be divided into three main types: (i) word meaning in context (Chapters 3, 4 and 6), (ii) polysemy-related properties (Chapters 5 and 7), and (iii) semantic relationships between words (Chapters 8 and 9). We provide here an overview of what we have learnt about different models.

i. We approximated word meaning in context using in-context lexical substitute and word similarity annotations. First, we evaluated different context-sensitive representations on the lexical substitution task (Chapter 3). Our results showed that models trained with a slot-filling objective, like context2vec and especially BERT, are more suitable for this task than a model focused on next word prediction, like ELMo. The same trend was observed for usage similarity estimation (Chapter 4), where contextualised BERT representations made high quality predictions. ELMo representations do not reflect usage similarity as well, although predictions improve slightly when incorporating representations from surrounding words in close proximity to the target, which are used for target word prediction during training. All contextualised representations give better usage similarity judgments than static representations. These results demonstrate their advantage in representing word meaning in context. However, in Chapter 6, we found that the similarity estimates derived from BERT representations are affected by sentence changes that do not alter the meaning of the sentence or of the words in it.

ii. The lexical properties investigated in this thesis are the partitionability of words into senses and their polysemy level. In Chapter 5, we extended past work on the first

property (McCarthy et al., 2016) that proposed to view partitionability as clusterability of a word's semantic space. We again found that BERT is able to make the highest quality predictions, outperforming ELMo, context2vec, and McCarthy et al.'s (2016) approach that relied on manual substitute annotations. These results are promising and suggest that the semantic space built by BERT reflects the different ambiguity types of words. Clusterability predictions, however, did not scale well on a larger corpus. We attributed this mainly to the quality of the sentences used (which were randomly selected and did not necessarily contain instances of all senses of a word) and to the model's high sensitivity to specific collocational and contextual differences in word usage. Our findings regarding the models' knowledge about polysemy presented in Chapter 7 are, however, highly interesting. The controlled sense distributions used in our experiments allowed us to conclude that the models, and particularly English BERT, encode information about words' number of senses that is acquired during pre-training, and which is present in the representations of new word instances regardless of their context.

iii. Finally, we investigated two other aspects of lexical meaning which are reflected in the relationships between words. First, we discovered that BERT representations encode rich knowledge about adjective intensity (Chapter 8) that is reflected in the similarity estimations obtained for scalar adjectives. We proposed a simple and resource-lean methodology that effectively uses representations for ranking adjectives by intensity. BERT representations also proved to be effective for distinguishing scalar from relational adjectives, although static embeddings obtained similarly good results on this task. Second, we investigated the knowledge that BERT contains about noun properties, as expressed in adjective-noun constructions. We found that it is hard for the model to make good predictions in an unsupervised cloze task setting, but that the knowledge can be learnt to some extent in a supervised binary classification task. We, however, prefer to be conservative in the strength of our conclusions since these results are strongly tied to the particular dataset and cloze prompts that were used. Nevertheless, our results show that the model can successfully leverage the knowledge that is relevant for detecting the entailment relationship between nouns and the AN constructions where they can appear when fine-tuned on a dataset specifically curated for this task.

Overall, we found contextual models, and in particular English BERT, to obtain unprecedented performance on lexical semantics tasks. Contextualised representations offer a great advantage over static methods and faithfully reflect different aspects of word meaning, even if they can be further enhanced with the integration of external knowledge.

Location of the knowledge In our experiments, we also investigated the location of different types of knowledge in terms of the model layers where these seem to be better encoded. A general trend observed with English BERT is that higher layers perform better at lexical semantics tasks. We observed this in our experiments on usage similarity estimation, clusterability and polysemy level prediction, scalar adjective ranking, and noun property prototypicality

detection. Previous work has shown that out-of-context (i.e. word type level) lexical knowledge is most prominent in lower layers (Vulić et al., 2020). Most of our experiments, however, involved in-context estimations, with the exception of settings where we aggregate information across word instances, as for polysemy level prediction and scalar adjective ranking (Chapters 7 and 8).

Multilinguality Although the bulk of our work addresses English, we also include experiments in other languages. Our word instance similarity prediction experiments (Chapter 6) involved fine-tuning models on Finnish data, while in our polysemy level prediction and scalar adjective ranking experiments we also addressed French, Spanish and Greek (Chapters 7 and 8). The trends observed and the results obtained with multilingual BERT and monolingual BERT models in these languages are somewhat different to those obtained with English BERT. One observation regarding multilingual BERT (common in Chapters 6 and 7) is that usage similarity estimates derived from the representations of this model are very high, even for different words, and fall in a very narrow range of values compared to language-specific models which give similarity values in a wider range. We concluded that mBERT has higher anisotropy, which means that its representations occupy a narrow cone in space. Additionally, mBERT and language-specific models tend to perform worse than English BERT on the lexical semantics tasks addressed in our work, and the best-performing layers vary across models. Semantic information does not seem to always be located in the upper half of the models as in English BERT.

Improving the quality of contextualised representations Throughout the thesis, we experimented with different ways for improving the lexical semantic knowledge encoded in the models and their representations. The two main strategies proposed have been the addition of training data and the use of manual and automatic substitute annotations. In Chapter 3, we used additional sentences to obtain representations of candidate substitutes. In Chapter 4, we proposed to use data manually annotated with substitutes to train models for usage similarity prediction, and incorporated features based on substitute overlap. In Chapter 5, we built representations based on automatic substitutes to predict word clusterability. These approaches, however, did not always have the desired effect, and resulted in no, or very slight, improvements in the corresponding tasks.

In Chapter 6, we combined the two strategies for in-context word similarity estimation. We fine-tuned BERT on a related task where the model has to learn to distinguish correct in-context substitutes for a target word from other synonyms and unrelated words. We collected data for this task using automatic substitute annotations. The proposed approach led to an improvement in performance compared to the BERT model without fine-tuning. This is encouraging, as one advantage of this approach is that it is possible to test it in other languages present in the Paraphrase Database (Ganitkevitch et al., 2013; Pavlick et al., 2015) (which we used as a pool of candidate substitutes) with no need for manual annotations. It shows that although the similarity estimates derived from BERT representations are of high quality, they

can be further enhanced using external knowledge, such as automatic substitute annotations.

10.2 Perspectives

The work presented in this thesis answers several questions about word meaning representation in neural language models, but also opens up exciting avenues worth exploring in the future that we discuss below.

Multilingual and language-specific BERT models Our study involving multilingual BERT and BERT in languages other than English (Finnish, French, Spanish and Greek) showed that these models do not perform as well on lexical semantics tasks as English BERT. For the multilingual model, this can be partly due to its higher anisotropy. As for monolingual models, however, there is no obvious reason for models in other languages to perform worse than English BERT. possible explanation for the worse results obtained in the languages studied in this thesis can be that they have a richer morphology and therefore need more training data. The reason for the lower performance could also lie in the quality of the datasets used for evaluation. For example, the EuroSense data (Delli Bovi et al., 2017) (used in Chapter 7) involve automatic annotations, which contain different amounts of noise in different languages, and the Finnish portion of CoSimLex (used in Chapter 6) has a limited size. A more thorough investigation of the quality of these datasets would be needed in order to disentangle this factor from factors related to the inner workings of the models or to specificities of each language. Another intriguing fact is that the self-similarity patterns observed throughout the different Transformer layers vary across models. This suggests that contextualisation (Ethayarajh, 2019) does not take place in the same way, and that information flows differently through layers in Transformer models for different languages. It would be interesting to investigate to what extent this is due to differences in model design, and why different languages and language combinations give rise to different self-similarity patterns.

Wordpiece handling Words have been the focus of this thesis, but BERT-like models, extensively used in our experiments, rely on a different kind of unit: wordpieces (Schuster and Nakajima, 2012; Wu et al., 2016), or more generally sub-word units (Sennrich et al., 2016). While many words in the vocabulary have a dedicated wordpiece, this is not the case for all words, which are sometimes split into multiple wordpieces. In most of our experiments, we have adopted a straightforward strategy to deal with these cases, which consists in averaging the representations of all wordpieces that form a word. We have reasons to believe, however, that the representations of words that are split into several wordpieces encode lower quality semantic information than those of words composed of a single wordpiece. First, the smaller pieces these words are made of are shared with other vocabulary items, and therefore encode information that is not exclusive to them. Second, we observe that models whose tokenisers tend to split words more often are not as good at discriminating words with different polysemy levels as well as the uncased English model (Chapter 7). This happened with cased BERT

(which has a smaller vocabulary) and with multilingual BERT. mBERT has a four times larger vocabulary than the uncased English BERT, but it needs to account for the vocabulary of about 100 more languages. In our scalar adjective ranking experiments, we tested the behaviour of the models when the last wordpiece was omitted. This yields better results than averaging all pieces in all the monolingual models tested, presumably because it removes pieces with morphological information that was not relevant for the task. However, a more systematic study is needed in order to understand the effect that word splitting has on the representations, and to find the best strategy for representing these words for different word-level tasks.

Other aspects of lexical meaning We have studied several aspects of lexical meaning, but the field of lexical semantics is vast and many interesting areas still remain unexplored. With respect to lexical ambiguity, for example, one aspect worth investigating is the representation of regular polysemy. It would be interesting to explore whether common patterns can be detected in the contextualised representations of words that present the same kind of alternations. This could, for example, be the case for words like *bottle* and *glass*, which express a CONTAINER-CONTENT alternation (e.g. "The *bottle* broke" vs "I drank the whole *bottle*").

As discussed at the end of Chapter 8, our methodology for detecting adjective intensity could serve to explore other connotational aspects of lexical meaning. For example, it could be used to investigate whether the representations encode information about the relative formality and complexity of near-synonyms. Other directions to pursue are emotion detection (Mohammad, 2018), the analysis of words that belong to the same scale with opposing polarity (e.g. *happy* and *sad*) and, more generally, the representation of antonyms. Finally, the effect of negation and of adverbial intensifiers on scalar adjectives' relative intensity (e.g. *not happy* and *very happy*) is also worth exploring. Negation and negated adjectives constitute challenges for distributional models (Aina et al., 2018), including BERT (Ettinger, 2020), while adverbial intensifiers (e.g. *very*, *quite*) tend to change the intensity of the word they modify (Cocos et al., 2018; Bostan and Klinger, 2019).

Improving representation quality We have shown that, in spite of the quality usage similarity predictions obtained with BERT representations, there is still room for improving their representation of word meaning. As discussed in Chapter 6, this is a research area that has drawn much attention in the last few years (Lauscher et al., 2019; Shi et al., 2019). In our experiments, we have fine-tuned BERT on data obtained with automatic substitute annotations and have shown that they can be helpful. However, there are other promising techniques that deserve further experimentation, such as the integration of token information at the embedding layer (Qu et al., 2019) and the combination of contextualised representations with static embeddings (Liu et al., 2020).

Model-agnostic methodology The methodology used in our analyses mainly relies on calculations involving representations in the vector space. As a consequence, it is generally model-agnostic, and can be applied to any kind of token-level vector representations. The

development of deep language models is currently a highly active area of research, with many new models being designed and released at a fast pace. We believe that our work can be useful for the analysis and comparison of other existing and future models. The study of models in languages other than English is, of course, restricted to the availability of evaluation datasets. Our contribution in this respect is the creation of a dataset in French, Spanish and Greek which will enable further research on scalar adjective representations in these languages.

Chapter A

Appendix

A.1 Word Usage Similarity Estimation

A.1.1 Substitute Filtering: Development Results

We report results of the different substitute filtering mechanisms described in Section 4.3.2.2 on the portion of LexSub data (McCarthy and Navigli, 2007) that does not contain Usim judgments (Erk et al., 2009, 2013). We measure the quality of the filtered substitutes against the gold standard annotations using F1-score and Precision. This is a way of considering both Precision and Recall, but giving more weight to Precision. We do this because we believe that, for the usage similarity estimation task, retaining substitutes that are correct is more important than retaining all the correct substitutes. Table A.1 shows results for annotations assigned by context2vec using each pool of substitutes (AUTO-LSCNC and AUTO-PPDB).

	AUTO-LSCNC			AUTO-PPDB		
Filter	**F1**	**Precision**	**Avg**	**F1**	**Precision**	**Avg**
Highest 10	0.332	0.224	0.278	0.245	0.162	0.204
Highest 5	**0.375**	0.305	0.340	**0.290**	0.234	0.262
PPDB	0.333	0.357	0.345	0.268	**0.269**	**0.269**
GloVe ($T = 0.1$)	0.371	0.325	0.348	0.266	0.222	0.244
GloVe ($T = 0.2$)	0.373	0.339	**0.356**	0.268	0.231	0.246
GloVe ($T = 0.3$)	0.353	**0.341**	0.347	0.266	0.250	0.258
No filter	0.248	0.152	0.200	0.142	0.080	0.111

Table A.1: Results of different substitute filtering strategies applied to annotations assigned by context2vec when using the LexSub/CoInCo pool of substitutes (AUTO-LSCNC) and the PPDB pool (AUTO-PPDB).

A.1.2 Feature Ablation on Usim

Results of the feature ablation experiments performed on the Usim development sets (described in Section 4.3.3) are given in Table A.2. For each word in Usim, we train models removing one feature at a time and collect their results on the development set. We report the average Spearman's ρ over all words for every model.

Ablation	**Gold**	AUTO-LSCNC	AUTO-PPDB
None	0.729	0.538	0.524
Substitute cosine similarity	0.701	0.537	0.524
Common substitutes	0.722	0.538	0.524
GAP	0.730	0.537	0.523
c2v	0.730	0.539	0.523
Bert avg (4) (tw)	0.700	0.348	0.283

Table A.2: Results of feature ablation experiments for systems trained on the Usim dataset using gold substitutes as well as automatic substitutes from different pools, Lexsub/CoInCo (AUTO-LSCNC) and PPDB (AUTO-PPDB). We report the average Spearman ρ correlation on the development sets across all target words. Rows indicate the feature that is removed each time. For BERT, *tw* means we use the representation of the target word.

A.1.3 Development Experiments on WiC 0.1

Table A.3 shows the accuracy of different configurations on the WiC development set. For ELMo, we used a context window (cw) of size 2 because it was shown to work better than the sentence embedding (cf. Section 4.6).

Training set	Features	Accuracy
WiC	BERT avg 4 (tw)	65.24
	c2v	57.69
	ELMo top |cw|=2	61.11
	USE	63.68
	SIF	60.97
	Substitute-based	55.41
	Embedding based	**67.95**
	Combined	**66.81**
WiC + CoInCo	BERT avg 4 (tw)	64.96
	c2v	58.12
	ELMo top |cw|=2	61.11
	USE	63.53
	SIF	59.97
	Substitute-based	56.13
	Embedding-based	**68.66**
	Combined	**66.81**

Table A.3: Accuracy of different features and feature combinations on the WiC development set. On this dataset, the two best types of embeddings, that were chosen for the Embedding-based and Combined configurations, were BERT and USE. The Substitute-based and Combined models both use features of automatically substitutes from the PPDB pool, and back off to the Embedding-based model when there were no paraphrases available for the target word in PPDB. For BERT, *tw* means we use the representation of the target word.

A.2 Word Sense Clusterability Estimation

A.2.1 Clusterability Results by Lemma

Table A.4 shows Usim words ranked by their clusterability according to Umid and Uiaa. We also include the ranking by SIL using BERT-AGG representations at the 10th layer (Section 5.4.)

by Umid		**by Uiaa**		**by SIL (BERT-AGG)**		**by Umid**		**by Uiaa**		**by SIL (BERT-AGG)**	
fresh.a	0.76	new.a	0.01	new.a	0.12	paper.n	0.44	ring.n	0.53	call.v	0.23
raw.a	0.73	suffer.v	0.04	hold.v	0.13	soft.a	0.44	shed.v	0.53	light.a	0.23
softly.r	0.73	function.n	0.11	suffer.v	0.14	flat.a	0.44	shade.n	0.55	post.n	0.25
strong.a	0.73	fresh.a	0.17	lead.n	0.15	rich.a	0.41	heavy.a	0.57	heavy.a	0.25
special.a	0.70	investigator.n	0.18	hard.r	0.15	figure.n	0.39	fix.v	0.59	rich.a	0.2
throw.v	0.70	field.n	0.25	function.n	0.15	account.n	0.39	match.n	0.59	check.v	0.26
hard.r	0.64	work.v	0.27	strong.a	0.15	skip.v	0.38	dry.a	0.59	right.r	0.26
work.v	0.64	raw.a	0.29	draw.v	0.16	charge.n	0.38	rude.a	0.61	tap.v	0.26
solid.a	0.63	neat.a	0.31	solid.a	0.16	dry.a	0.38	paper.n	0.63	poor.a	0.26
function.n	0.62	strong.a	0.31	field.n	0.17	light.a	0.36	clear.v	0.63	shed.v	0.27
put.v	0.62	throw.v	0.32	ring.n	0.17	rough.a	0.35	rough.a	0.63	severely.r	0.27
dismiss.v	0.61	put.v	0.34	neat.a	0.18	investigator.n	0.35	order.v	0.64	skip.v	0.27
heavy.a	0.60	hard.r	0.34	work.v	0.18	range.n	0.34	call.v	0.65	put.v	0.27
neat.a	0.58	bar.n	0.35	fresh.a	0.18	poor.a	0.34	right.r	0.65	figure.n	0.27
bright.a	0.55	check.v	0.35	bar.n	0.18	fix.v	0.34	account.n	0.66	investigator.n	0.28
rude.a	0.53	scrap.n	0.36	raw.a	0.19	order.v	0.33	bright.a	0.67	paper.n	0.28
draw.v	0.53	special.a	0.37	stiff.a	0.19	match.n	0.33	charge.v	0.68	bright.a	0.28
check.v	0.52	stiff.a	0.40	soft.a	0.10	ring.n	0.33	post.n	0.69	execution.n	0.28
scrap.n	0.51	poor.a	0.43	clear.v	0.20	severely	0.33	tap.v	0.70	flat.a	0.29
shed.v	0.49	hold.v	0.47	rough.a	0.21	suffer.v	0.32	skip.v	0.70	match.n	0.30
lead.n	0.49	lead.n	0.47	rude.a	0.21	shade.n	0.30	rich.a	0.73	shade.n	0.31
right.r	0.48	softly.r	0.48	throw.v	0.21	bar.n	0.30	range.n	0.74	charge.v	0.32
hold.v	0.48	light.a	0.49	dismiss.v	0.21	coach.n	0.27	coach.n	0.74	coach.n	0.33
field.n	0.47	solid.a	0.49	scrap.v	0.21	charge.v	0.24	execution.n	0.78	range.n	0.33
stiff.a	0.46	draw.v	0.50	dry.a	0.21	new.a	0.23	severely.r	0.78	fix.v	0.35
execution.n	0.46	figure.n	0.50	special.a	0.22	post.n	0.22	charge.n	0.81	account.n	0.37
clear.v	0.45	soft.a	0.51	order.v	0.23	call.v	0.18	flat.a	0.85	charge.n	0.41
tap.v	0.45	dismiss.v	0.52	softly.r	0.23	fire.v	0.17	fire.v	0.93	fire.v	0.44

Table A.4: Ranking of lemmas from less to more clusterable by the gold-standards and by the clusterability estimations obtained with the best model (BERT-AGG, 10th layer, SIL metric).

A.3 Polysemy Level Prediction

A.3.1 Complete `poly-same` and `poly-bal` Results

In Figures A.1 and A.2 we report the average *SelfSim* obtained with BERT models for the different `poly` bands in the `poly-same` and `poly-bal` sentence pools, respectively (Section 7.3.1). Figure A.3 contains the same information for the ELMo model.

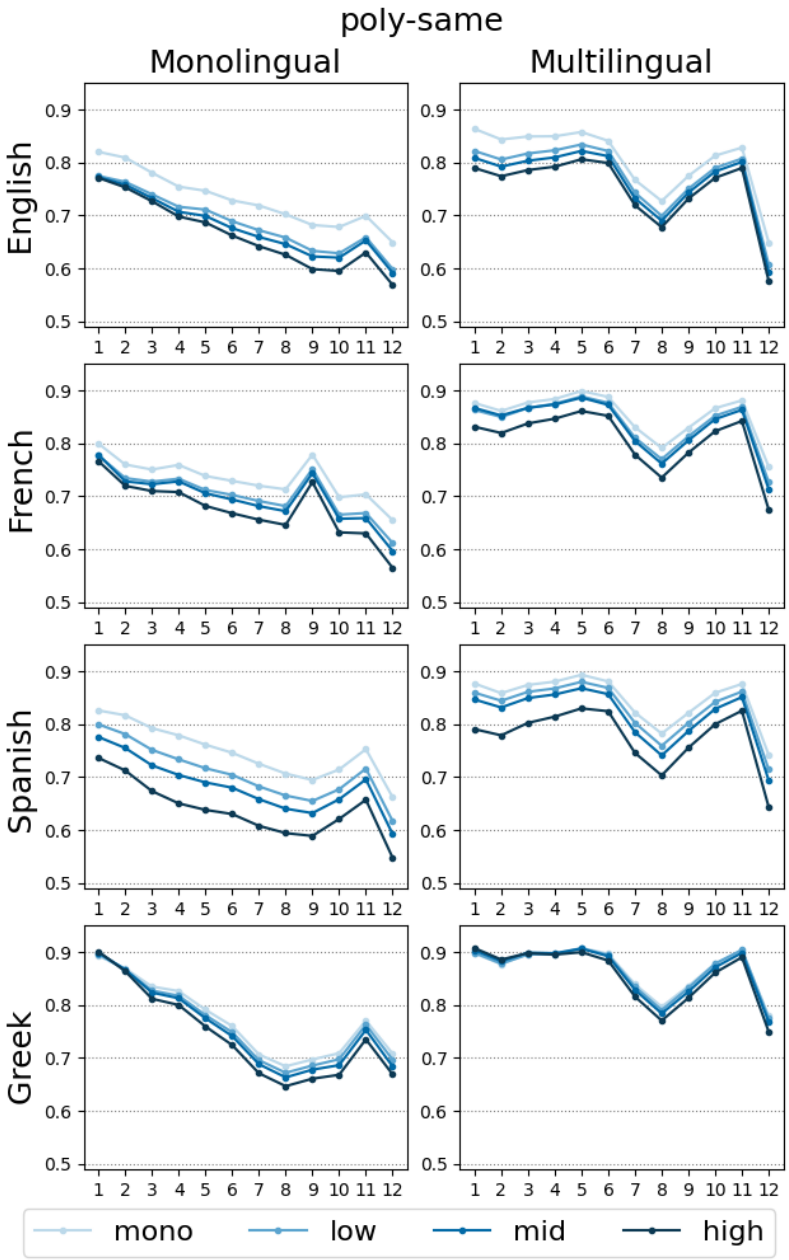


Figure A.1: Average *SelfSim* obtained with monolingual BERT models (left column) and mBERT (right column) in all languages for `mono` and `poly` lemmas in different polysemy bands in the `poly-same` sentence pool.

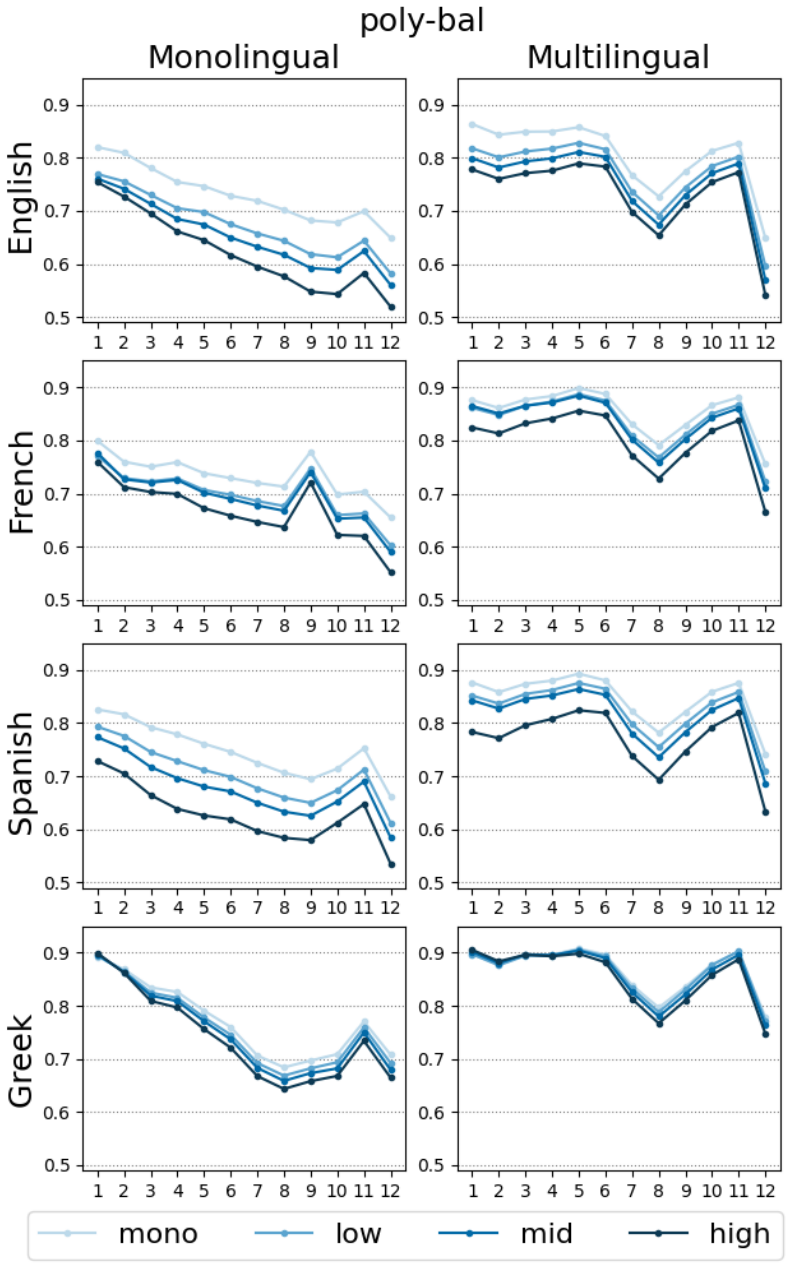


Figure A.2: Average *SelfSim* obtained with monolingual BERT models (left column) and mBERT (right column) in all languages for `mono` and `poly` lemmas in different polysemy bands in the `poly-bal` sentence pool.

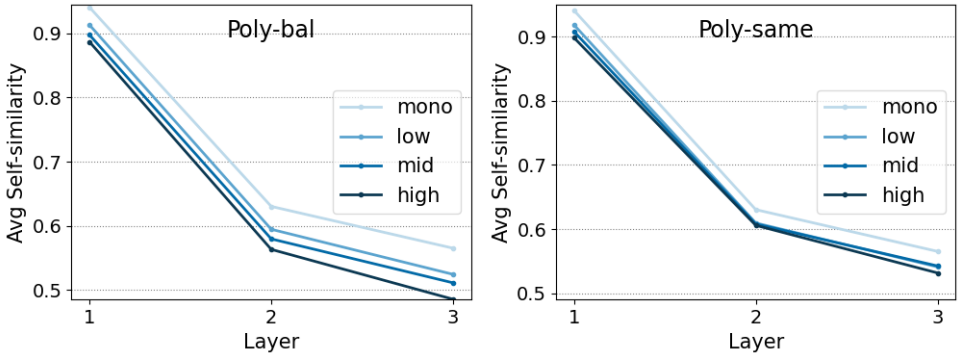


Figure A.3: Average *SelfSim* obtained with **ELMo** representations for `mono` and `poly` lemmas in different polysemy bands in the `poly-same` and `poly-bal` sentence pools.

A.3.2 Controlling for Frequency and PoS: mBERT Results

Figure A.4 contains the average *SelfSim* obtained in the FREQ-bal and POS-bal bands with mBERT (Section 7.4.3).

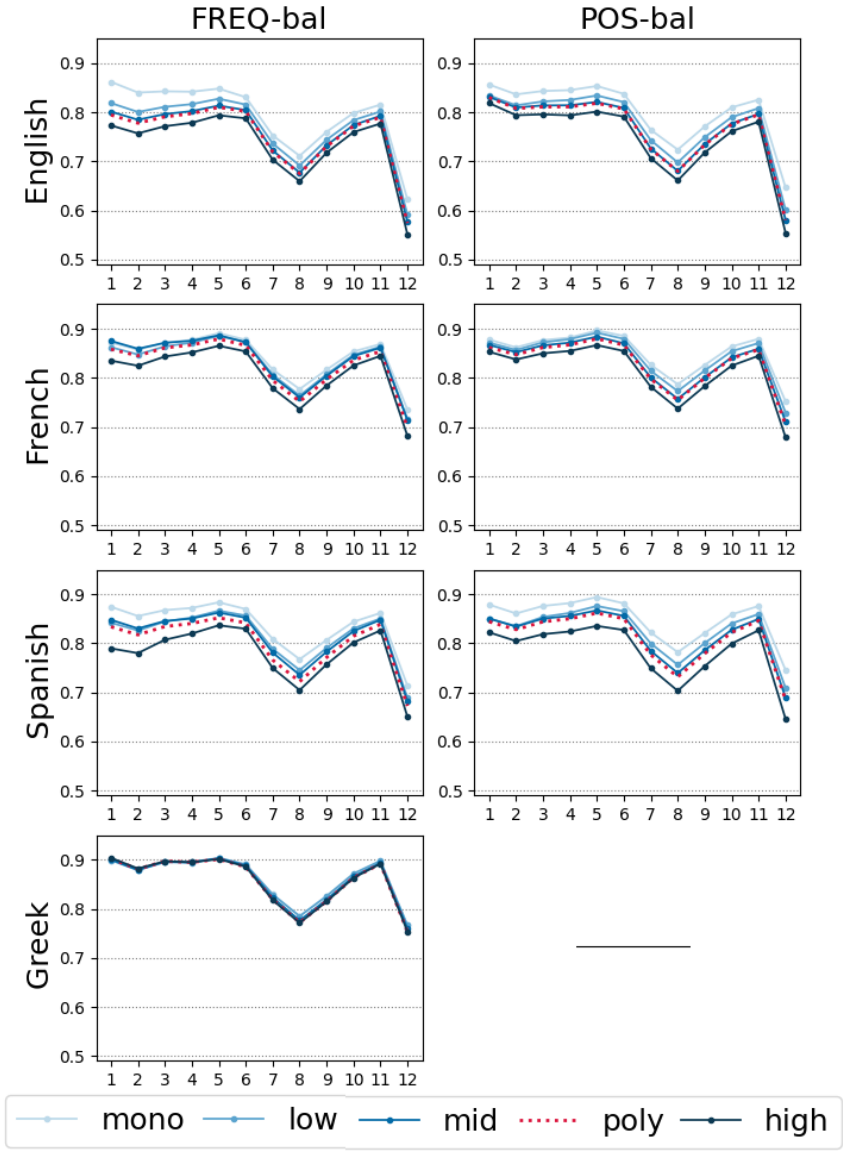


Figure A.4: Average *SelfSim* inside the `poly` bands balanced for frequency (FREQ-bal) and part of speech (POS-bal). *SelfSim* is calculated using representations generated by **mBERT** from sentences in each language-specific pool. We do not balance the Greek dataset for PoS because it only contains nouns.

A.4 Scalar Adjective Ranking

A.4.1 Hearst Patterns

Figure A.5 illustrates the dependency structure of the following Hearst patterns, used to filter out sentences containing scalar adjectives (Section 8.2.2). We remove these sentences from our ukWaC and Flickr datasets.[1].

- *[NP] and other [NP]*
- *[NP] or other [NP]*
- *[NP] such as [NP]*
- *Such [NP] as [NP]*
- *[NP], including [NP]*
- *[NP], especially [NP]*

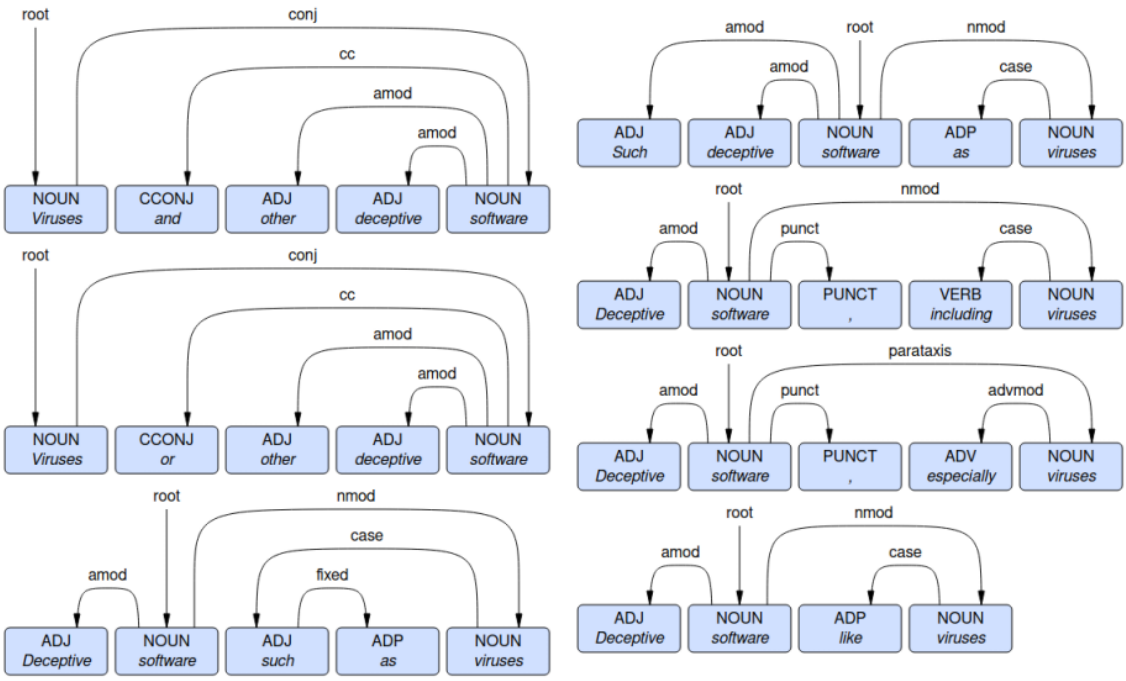


Figure A.5: Dependency structure of Hearst patterns.

[1]Graphs in Figure A.5 were created with the visualisation tool available at https://urd2.let.rug.nl/~kleiweg/conllu/

A.4.2 Evaluation of Sentence Selection Methods

To identify the most appropriate method for selecting sentences where all adjectives in a scale fit, we use data from the Concepts in Context (CoInCo) corpus (Kremer et al., 2014). We collect instances of adjectives, nouns and verbs in their base form.[2] For a word w, we form instance pairs (w_i-w_j with $i \neq j$) with similar meaning as reflected in their shared substitutes. We allow for up to two unique substitutes per instance, which we assign to the other instance in the pair with zero frequency. We keep instances with n substitutes, where $2 \leq n \leq 8$ (the lowest and highest number of adjectives in a scale). This results in 5,954 pairs.

We measure the variation in an instance pair in terms of substitutes and their frequency scores using the *coefficient of variation* (VAR). VAR is the ratio of the standard deviation to the mean and is, therefore, independent from the unit used. A higher VAR indicates that not all substitutes are good choices in a context. We keep the 500 pairs with the highest VAR difference, where one sentence is a better fit for all substitutes than the other. For example, *private*, *individual* and *person* were proposed as substitutes for *personal* in "personal insurance lines", but *private* was the preferred choice for "personal reasons". The tested methods must identify which sentence in a pair is a better fit for all substitutes.

For sentence selection, we experiment with the three fluency calculation methods presented in Section 8.2.2: BERTPROB (the BERT probability of each substitute to be used in the place of the [MASK] token); BERTPPX (the perplexity assigned by BERT to the sentence generated through substitution); and CONTEXT2VEC (the cosine similarity between the context2vec representations of a substitute and the context).

We also test VAR and standard deviation (STD) as metrics for measuring variation in the fluency scores assigned to a sentence pair by the three methods. We evaluate the sentence selection methods and variation metrics on the 500 pairs retained from CoInCo. We report their accuracy, calculated as the proportion of pairs where a method correctly guesses the instance with the lowest variation in a pair. We compare results to those of a baseline that always proposes the first instance in a pair. The results in Table A.5 show that the task is difficult for all methods. Their accuracy is slightly higher than the baseline accuracy, which outperforms BERTPROB with VAR. We use the best combination (CONTEXT2VEC with STD) to select sentences for our experiments.

Method	Variation Metric	Accuracy
BERTPROB	STD	0.524
	VAR	0.488
BERTPPX	STD	0.518
	VAR	0.536
context2vec	STD	**0.594**
	VAR	0.588
1st sentence Baseline		0.506

Table A.5: Accuracy of the three fluency calculation methods on the 500 sentence pairs collected from CoInCo. Comparison to a first sentence baseline.

[2]This filtering serves to control for morphological variation which could result in unnatural substitutions since CoInCo substitutes are in lemma form.

A.4.3 Adjustment for Ties

Table A.6 contains results of the DIFFVEC method described in Section 8.2.4 with the adjustment for ties. For two adjacent adjectives (a_i, a_j) in the ranking proposed by DIFFVEC, we check if their cosine similarities to $\overrightarrow{dVec}$ are very close ($diffsim = \text{sim}(\overrightarrow{dVec}, \overrightarrow{a_i})$ - $\text{sim}(\overrightarrow{dVec}, \overrightarrow{a_j})$. If $|diffsim|$ (the absolute value of $diffsim$) < 0.01, we count them as a tie, meaning that a_i and a_j are considered to be situated at the same intensity level. Note that this procedure may give different results when the pairwise comparison starts at different ends of the proposed ranking. We establish ties starting from the a with lowest intensity in the ranking proposed by DIFFVEC.

			DEMELO (DM)			CROWD (CD)			WILKINSON (WK)		
		Method	P-ACC	τ	ρ_{avg}	P-ACC	τ	ρ_{avg}	P-ACC	τ	ρ_{avg}
BERT	**ukWaC**	DIFFVEC-DM	-	-	-	0.733_{8}	0.673_{8}	$\mathbf{0.749}_{12}$	0.885_{6}	0.830_{11}	0.826_{6}
		DIFFVEC-CD	$\mathbf{0.644}_{8}$	$\mathbf{0.452}_{8}$	$\mathbf{0.518}_{8}$	-	-	-	0.820_{10}	0.721_{11}	0.780_{11}
		DIFFVEC-WK	0.546_{6}	0.295_{6}	0.324_{6}	0.721_{7}	0.627_{10}	0.698_{10}	-	-	-
	Flickr	DIFFVEC-DM	-	-	-	0.746_{12}	$\mathbf{0.685}_{12}$	0.718_{8}	$\mathbf{0.902}_{9}$	$\mathbf{0.851}_{9}$	$\mathbf{0.871}_{4}$
		DIFFVEC-CD	0.605_{11}	0.388_{11}	0.465_{11}	-	-	-	0.836_{8}	0.746_{7}	0.762_{7}
		DIFFVEC-WK	0.541_{2}	0.296_{1}	0.299_{1}	0.702_{8}	0.647_{8}	0.710_{8}	-	-	-
	Random	DIFFVEC-DM	-	-	-	$\mathbf{0.724}_{9}$	0.652_{9}	0.719_{8}	0.885_{11}	0.818_{6}	0.833_{10}
		DIFFVEC-CD	0.619_{8}	0.412_{8}	0.488_{8}	-	-	-	0.819_{12}	0.765_{10}	0.833_{10}
		DIFFVEC-WK	0.522_{2}	0.251_{6}	0.285_{6}	0.712_{10}	0.614_{9}	0.680_{9}	-	-	-
word2vec		DIFFVEC-DM	-	-	-	0.648	0.508	0.550	0.754	0.583	0.655
		DIFFVEC-CD	0.604	0.403	0.446	-	-	-	0.803	0.656	0.661
		DIFFVEC-WK	0.568	0.329	0.402	0.606	0.414	0.445	-	-	-

Table A.6: Results of our DIFFVEC adjective ranking method on the DEMELO, CROWD and WILKINSON datasets with the adjustment for ties. We report results with contextualised (BERT) representations obtained from different SENT-SETs (ukWaC, Flickr, Random) and with static (word2vec) vectors.

A.4.4 DIFFVEC with a Single Sentence

Table A.7 contains results for DIFFVEC-1 (+)/(−) and DIFFVEC-5 when using a single sentence for building $\overrightarrow{dVec}$.

			DEMELO			CROWD		
		# Scales	P-ACC	τ	ρ_{avg}	P-ACC	τ	ρ_{avg}
BERT	**ukWaC**	1 (+)	0.651_{10}	0.433_{10}	0.501_{10}	0.682_{10}	0.553_{10}	0.622_{7}
		1 (−)	0.597_{1}	0.315_{1}	0.352_{1}	0.639_{12}	0.458_{12}	0.543_{12}
		5	$\mathbf{0.655}_{7}$	$\mathbf{0.443}_{7}$	$\mathbf{0.530}_{7}$	$\mathbf{0.691}_{11}$	$\mathbf{0.575}_{11}$	$\mathbf{0.675}_{11}$
	Flickr	1 (+)	0.639_{9}	0.410_{9}	0.432_{9}	0.676_{8}	0.550_{8}	0.604_{8}
		1 (−)	0.602_{3}	0.329_{3}	0.372_{3}	0.629_{4}	0.443_{4}	0.479_{4}
		5	0.624_{11}	0.380_{11}	0.452_{11}	0.683_{11}	0.562_{11}	0.606_{12}
	Random	1 (+)	0.631_{11}	0.401_{11}	0.451_{11}	0.676_{8}	0.536_{8}	0.589_{8}
		1 (−)	0.611_{9}	0.356_{9}	0.444_{9}	0.648_{11}	0.479_{11}	0.500_{11}
		5	0.622_{4}	0.371_{4}	0.417_{3}	0.685_{7}	0.559_{7}	0.588_{7}
word2vec		1 (+)	0.602	0.334	0.364	0.624	0.419	0.479
		1 (−)	0.613	0.359	0.412	0.661	0.506	0.559
		5	0.641	0.415	0.438	0.688	0.559	0.601

Table A.7: Results of DIFFVEC using a single positive (1 (+)) or negative (1 (−)) adjective pair, and five pairs (5). These are results obtained with a $\overrightarrow{dVec}$ built from only one sentence (instead of ten as in Table 8.5).

A.4.5 Comparison of Wordpiece Selection Methods

Table 8.9 in Chapter 8 contains results of the DIFFVEC method with the best approach for selecting wordpieces (WPs) for each model. In Table A.8, we present results obtained using the alternative approach for each model and language:

- for all monolingual models and the multilingual model for English, Table A.8 contains results obtained with the WP approach;
- for the multilingual models in the other languages, we show results with WP-1.

The best approach was determined by comparing their average scores across the different methods. Some configurations improve, but they yield overall worse results per model, especially in Spanish. Differences between WP and WP-1 are generally more pronounced in the multilingual models than in the monolingual models.

		EN			FR			ES			EL		
		Mono WP			**Mono WP**			**Mono WP**			**Mono WP**		
		P-ACC	τ	ρ_{avg}	P-ACC	τ	ρ_{avg}	P-ACC	τ	ρ_{avg}	P-ACC	τ	ρ_{avg}
DM	DV-1 (+)	**.664**$_9$	**.463**$_9$	**.531**$_9$	**.617**$_3$	**.384**$_3$	**.406**$_3$	**.652**$_9$	**.367**$_9$	**.390**$_9$	.546$_8$	.201$_8$	.215$_8$
	DV-WK	.557$_9$	.246$_9$	.284$_6$	.517$_1$	.170$_1$	.140$_1$	.645$_{10}$	.353$_{10}$	.313$_{10}$	**.557**$_2$	**.226**$_2$	**.240**$_2$
WK	DV-1 (+)	.852$_7$	.705$_7$	.766$_1$	.612$_7$	.262$_1$	.215$_6$	**.763**$_8$	**.525**$_8$	**.755**$_6$	.632$_8$	.312$_8$	.256$_8$
	DV-DM	**.918**$_6$	**.836**$_6$	**.839**$_6$	.627$_2$	.292$_2$	.392$_2$	.746$_6$	.492$_6$	.658$_6$	**.779**$_1$1	**.617**$_{11}$	**.663**$_{11}$
		Multi WP			**Multi WP-1**			**Multi WP-1**			**Multi (unc) WP-1**		
DM	DV-1 (+)	.588$_4$	.301$_4$	.312$_4$	.549$_7$	.239$_7$	.276$_7$	.589$_3$	.229$_3$	.234$_1$	.524$_9$	.153$_9$	.171$_9$
	DV-WK	.516$_5$	.153$_{11}$	.198$_5$	.490$_2$	.113$_2$	.134$_7$	.603$_{12}$	.268$_{12}$	.287$_{12}$	.521$_6$	.146$_6$	.186$_6$
WK	DV-1 (+)	.820$_7$	.639$_7$	.667$_3$	.612$_3$	.262$_3$	.362$_3$	.746$_4$	.492$_4$	.608$_4$	.647$_9$	.358$_9$	.369$_9$
	DV-DM	.885$_7$	.770$_7$	.834$_7$	**.687**$_7$	**.412**$_7$	**.435**$_3$	.661$_{10}$	.322$_{10}$	.447$_6$	.662$_6$	.388$_6$	.444$_6$

Table A.8: Results of DIFFVEC (DV) methods with contextualised representations derived from monolingual and multilingual models for each language, using an alternative approach to selecting wordpieces (WP, WP-1) than the one used for the results reported in Table 8.9 in Chapter 8. For all languages but Greek, the multilingual model is cased.

A.5 Nouns' Semantic Properties and their Prototypicality

A.5.1 Properties Masking Results

Figure A.6 shows the average recall at positions @1, @5 and @10 of the ranked BERT-`base` and `large` predictions, when using sentences constructed with the templates that correspond to the labels on the x axis. Average is calculated over the words for which at least one correct attribute is found at the specific rank, as shown in Figure 9.1.

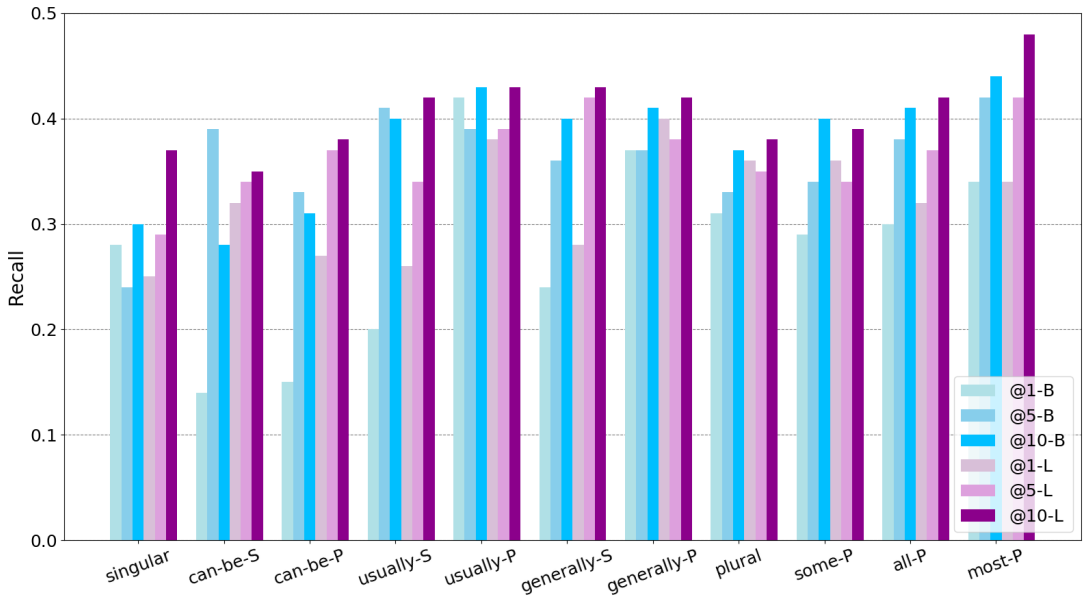


Figure A.6: Average recall at positions @1, @5 and @10 of the ranked BERT-`base` (B) and `large` (L) predictions for the cloze task addressing MRD attributes.

A.5.2 Detailed Embedding-based Classification results

Table A.9 lists the best configurations per $\overrightarrow{AN}$ type and per type of composition obtained with BERT embedding-based classification models on the HVD development set described in Section 9.4.

Figure A.7 shows the highest average accuracy obtained by each BERT layer on the HVD development set in these experiments.

$\overrightarrow{AN}$ **type**	**Composition**	**Layer**	**Similarity**
Ns_N, Ns_{AN}	addition	12	cosine & euclidean $(Ns_N, Ns_N + Ns_{AN})$
As_N	-	8	-
Ns_N, As_{AN}	difference	12	-
Ns_{AN}, As_{AN}	difference	12	-
Ns_{AN}	-	11	cosine (Ns_N, Ns_{AN})
Composition type	$\overrightarrow{AN}$ **type**	**Layer**	**Similarity**
addition	Ns_N, Ns_{AN}	12	cosine & euclidean $(Ns_N, Ns_N + Ns_{AN})$
difference	Ns_N, As_{AN}	12	-
concatenation	Ns_{AN}, As_{AN}	7	-
average	Ns_N, As_{AN}	5	-
multiplication	Ns_N, As_{AN}	7	euclidean $(Ns_N, Ns_N \odot Ns_{AN})$

Table A.9: Complete best configurations for every type of $\overrightarrow{AN}$ (top) and composition operations (bottom) with BERT embedding-based classifiers on the HVD development set.

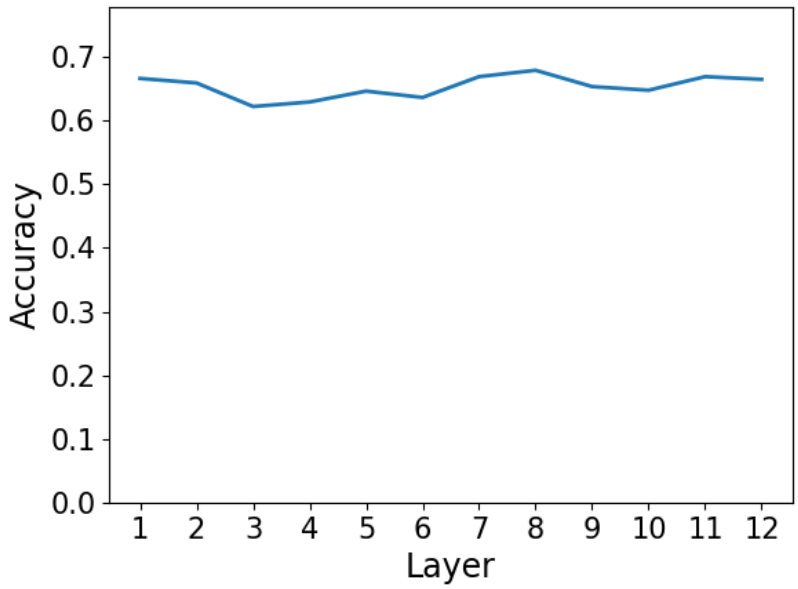


Figure A.7: Highest average accuracy obtained by the embedding-based classifier on the HVD development set at every BERT layer.

Bibliography

Margareta Ackerman and Shai Ben-David. 2009. Clusterability: A Theoretical Study. *Journal of Machine Learning Research*, 5:1–8.

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *Proceedings of 5th ICLR International Conference on Learning Representations*, Toulon, France.

Andreas Adolfsson, Margareta Ackerman, and Naomi C Brownstein. 2019. To cluster, or not to cluster: An analysis of clusterability methods. *Pattern Recognition*, 88:13–26.

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27, Boulder, Colorado. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

Eneko Agirre and David Martinez. 2004. Unsupervised WSD based on Automatically Retrieved Examples: The Importance of Bias. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 25–32, Barcelona, Spain. Association for Computational Linguistics.

Laura Aina, Raffaella Bernardi, and Raquel Fernández. 2018. A distributional study of negated adjectives and antonyms. In *Proceedings of the Fifth Italian Conference on Computational Lin-*

guistics (CLiC-it 2018), Torino, Italy, December 10-12, 2018, volume 2253 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Laura Aina, Kristina Gulordava, and Gemma Boleda. 2019. Putting Words in Context: LSTM Language Models and Lexical Ambiguity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3342–3348, Florence, Italy. Association for Computational Linguistics.

Alan Akbik, Tanja Bergmann, and Roland Vollgraf. 2019. Pooled Contextualized Embeddings for Named Entity Recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 724–728, Minneapolis, Minnesota. Association for Computational Linguistics.

Domagoj Alagić, Jan Šnajder, and Sebastian Padó. 2018. Leveraging Lexical Substitutes for Unsupervised Word Sense Induction. In *Proceedings of AAAI*, New Orleans, LA.

Marianna Apidianaki. 2009. Data-Driven Semantic Analysis for Multilingual WSD and Lexical Selection in Translation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 77–85, Athens, Greece. Association for Computational Linguistics.

Marianna Apidianaki. 2016. Vector-space models for PPDB paraphrase ranking in context. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2028–2034, Austin, Texas. Association for Computational Linguistics.

Marianna Apidianaki, Guillaume Wisniewski, Anne Cocos, and Chris Callison-Burch. 2018. Automated Paraphrase Lattice Creation for HyTER Machine Translation Evaluation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 480–485, New Orleans, Louisiana. Association for Computational Linguistics.

Yuki Arase and Jun'ichi Tsujii. 2019. Transfer Fine-Tuning: A BERT Case Study. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5393–5404, Hong Kong, China. Association for Computational Linguistics.

Nikolay Arefyev, Boris Sheludko, Alexander Podolskiy, and Alexander Panchenko. 2020. Always Keep your Target in Mind: Studying Semantics and Improving Performance of Neural Lexical Substitution. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1242–1255, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Carlos S. Armendariz, Matthew Purver, Matej Ulčar, Senja Pollak, Nikola Ljubešić, and Mark Granroth-Wilding. 2020a. CoSimLex: A Resource for Evaluating Graded Word Similarity in Context. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5878–5886, Marseille, France. European Language Resources Association.

Carlos Santos Armendariz, Matthew Purver, Senja Pollak, Nikola Ljubešić, Matej Ulčar, Ivan Vulić, and Mohammad Taher Pilehvar. 2020b. SemEval-2020 Task 3: Graded Word Similarity in Context. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 36–49, Barcelona (online). International Committee for Computational Linguistics.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. In *International Conference on Learning Representations (ICLR)*, Toulon, France.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Vancouver, Canada. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *Proceedings of the Sixth International Conference on Learning Representations*.

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with Bilingual Parallel Corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 597–604, Ann Arbor, Michigan. Association for Computational Linguistics.

Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. 2012. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 23–32, Avignon, France. Association for Computational Linguistics.

Marco Baroni, Raffaella Bernardi, and Roberto Zamparelli. 2014a. Frege in Space: A Program for Composition Distributional Semantics. In *Linguistic Issues in Language Technology, Volume 9, 2014 - Perspectives on Semantic Representations for Textual Inference*. CSLI Publications.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. *Journal of Language Resources and Evaluation*, 43(3):209–226.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014b. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland. Association for Computational Linguistics.

Marco Baroni and Alessandro Lenci. 2010. Distributional Memory: A General Framework for Corpus-Based Semantics. *Computational Linguistics*, 36(4):673–721.

Marco Baroni and Roberto Zamparelli. 2010. Nouns are Vectors, Adjectives are Matrices: Representing Adjective-Noun Constructions in Semantic Space. In *Proceedings of the 2010*

Conference on Empirical Methods in Natural Language Processing, pages 1183–1193, Cambridge, MA. Association for Computational Linguistics.

Sergey Bartunov, Dmitry Kondrashkin, Anton Osokin, and Dmitry Vetrov. 2016. Breaking Sticks and Ambiguities with Adaptive Skip-gram. In *artificial intelligence and statistics*, pages 130–138.

Yonatan Belinkov and James Glass. 2019. Analysis Methods in Neural Language Processing: A Survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300.

Chris Biemann. 2013. Creating a system for lexical substitutions from scratch using crowdsourcing. *Language Resources and Evaluation*, 47(1):97–122.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc., Beijing.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. In *Advances in Neural Information Processing Systems 29*, pages 4349–4357. Barcelona, Spain.

Rishi Bommasani, Kelly Davis, and Claire Cardie. 2020. Interpreting Pretrained Contextualized Representations via Reductions to Static Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4758–4781, Online. Association for Computational Linguistics.

Francis Bond and Kyonghee Paik. 2012. A survey of wordnets and their licenses. *Small*, 8(4):5.

Laura Ana Maria Bostan and Roman Klinger. 2019. Exploring fine-tuned embeddings that model intensifiers for emotion analysis. In *Proceedings of the Tenth Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA@NAACL-HLT 2019, Minneapolis, USA, June 6, 2019*, pages 25–34. Association for Computational Linguistics.

Zied Bouraoui, Jose Camacho-Collados, and Steven Schockaert. 2020. Inducing relational knowledge from bert. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7456–7463.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram Version 1. In *LDC2006T13*, Philadelphia, Pennsylvania. Linguistic Data Consortium.

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational linguistics*, 32(1):13–47.

John A Bullinaria and Joseph P Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior research methods*, 39(3):510–526.

José Camacho-Collados and Roberto Navigli. 2016. Find the word that does not belong: A framework for an intrinsic evaluation of word vector representations. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 43–50.

José Camacho-Collados and Mohammad Taher Pilehvar. 2018. From Word To Sense Embeddings: A Survey on Vector Representations of Meaning. *Journal of Artificial Intelligence Research*, 63:743–788.

José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. NASARI: a Novel Approach to a Semantically-Aware Representation of Items. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 567–577, Denver, Colorado. Association for Computational Linguistics.

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, and Jorge Pérez. 2020. Spanish Pre-Trained BERT Model and Evaluation Data. In *PML4DC at ICLR 2020*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1025–1035.

Billy Chiu, Anna Korhonen, and Sampo Pyysalo. 2016. Intrinsic Evaluation of Word Vectors Fails to Predict Extrinsic Performance. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 1–6, Berlin, Germany. Association for Computational Linguistics.

Gabriella Chronis and Katrin Erk. 2020. When is a bishop not like a rook? When it's like a rabbi! Multi-prototype BERT embeddings for estimating semantic relationships. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 227–244, Online. Association for Computational Linguistics.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What Does BERT Look at? An Analysis of BERT's Attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Anne Cocos and Chris Callison-Burch. 2016. Clustering Paraphrases by Word Sense. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1463–1472, San Diego, California. Association for Computational Linguistics.

Anne Cocos and Chris Callison-Burch. 2019. Paraphrase-Sense-Tagged Sentences. *Transactions of the Association for Computational Linguistics*, 7:714–728.

Anne Cocos, Skyler Wharton, Ellie Pavlick, Marianna Apidianaki, and Chris Callison-Burch. 2018. Learning Scalar Adjective Intensity from Paraphrases. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1752–1762, Brussels, Belgium. Association for Computational Linguistics.

Ronan Collobert and Jason Weston. 2008. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual Language Model Pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Mathias Creutz. 2018. Open Subtitles Paraphrase Corpus for Six Languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

D Alan Cruse. 1986. *Lexical semantics*. Cambridge university press.

Tim Van de Cruys and Marianna Apidianaki. 2011. Latent semantic word sense induction and disambiguation. In *ACL HLT 2011-49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1476–1485.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, MLCW'05, page 177–190, Berlin, Heidelberg. Springer-Verlag.

Claudio Delli Bovi, Jose Camacho-Collados, Alessandro Raganato, and Roberto Navigli. 2017. EuroSense: Automatic Harvesting of Multilingual Sense Annotations from Parallel Text. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 594–600, Vancouver, Canada. Association for Computational Linguistics.

Magdalena Derwojedowa, Maciej Piasecki, Stanislaw Szpakowicz, Magdalena Zawislawska, and Bartosz Broda. 2008. Words, Concepts and Relations in the Construction of the Polish WordNet. *Global WordNet Conference*, pages 162–177.

Sunipa Dev and Jeff M Phillips. 2019. Attenuating Bias in Word Vectors. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, Naha, Okinawa, Japan.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 350–356, Geneva, Switzerland. COLING.

William B. Dolan. 1994. Word Sense Ambiguation: Clustering Related Senses. In *COLING 1994 Volume 2: The 15th International Conference on Computational Linguistics*.

Katrin Erk, Diana McCarthy, and Nicholas Gaylord. 2009. Investigations on Word Senses and Word Usages. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 10–18, Suntec, Singapore. Association for Computational Linguistics.

Katrin Erk, Diana McCarthy, and Nicholas Gaylord. 2013. Measuring Word Meaning in Context. *Computational Linguistics*, 39(3):511–554.

Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 897–906, Honolulu, Hawaii. Association for Computational Linguistics.

Katrin Erk and Sebastian Padó. 2010. Exemplar-Based Models for Word Meaning in Context. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 92–97, Uppsala, Sweden. Association for Computational Linguistics.

Luis Espinosa-Anke, Thierry Declerck, Dagmar Gromann, Jose Camacho-Collados, and Mohammad Taher Pilehvar, editors. 2019. *Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5)*. Association for Computational Linguistics, Macau, China.

Kawin Ethayarajh. 2019. How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

Allyson Ettinger. 2020. What BERT Is Not: Lessons from a New Suite of Psycholinguistic Diagnostics for Language Models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. 2016. Probing for semantic evidence of composition by means of simple classification tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 134–139, Berlin, Germany. Association for Computational Linguistics.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting Word Vectors to Semantic Lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, Colorado. Association for Computational Linguistics.

Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems With Evaluation of Word Embeddings Using Word Similarity Tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 30–35, Berlin, Germany. Association for Computational Linguistics.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press, Cambridge, MA.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing Search in Context: The Concept Revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414.

John R Firth. 1957. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*.

Maxwell Forbes, Ari Holtzman, and Yejin Choi. 2019. Do Neural Language Representations Learn Physical Commonsense? *Proceedings of the 41st Annual Conference of the Cognitive Science Society*.

Juri Ganitkevitch and Chris Callison-Burch. 2014. The Multilingual Paraphrase Database. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4276–4283, Reykjavik, Iceland. European Language Resources Association (ELRA).

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The Paraphrase Database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, Atlanta, Georgia. Association for Computational Linguistics.

Aina Garí Soler and Marianna Apidianaki. 2020a. BERT Knows Punta Cana is not just beautiful, it's gorgeous: Ranking Scalar Adjectives with Contextualised Representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7371–7385, Online. Association for Computational Linguistics.

Aina Garí Soler and Marianna Apidianaki. 2020b. MULTISEM at SemEval-2020 Task 3: Fine-tuning BERT for Lexical Meaning. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 158–165, Barcelona (online). International Committee for Computational Linguistics.

Aina Garí Soler and Marianna Apidianaki. 2021a. Let's Play Mono-Poly: BERT Can Reveal Words' Polysemy Level and Partitionability into Senses. *To appear in Transactions of the Association for Computational Linguistics*.

Aina Garí Soler and Marianna Apidianaki. 2021b. Scalar Adjective Identification and Multilingual Ranking. In *To appear in Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Mexico City, Mexico. Association for Computational Linguistics.

Aina Garí Soler, Marianna Apidianaki, and Alexandre Allauzen. 2019a. LIMSI-MULTISEM at the IJCAI SemDeep-5 WiC Challenge: Context Representations for Word Usage Similarity Estimation. In *Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5)*, pages 6–11, Macau, China. Association for Computational Linguistics.

Aina Garí Soler, Marianna Apidianaki, and Alexandre Allauzen. 2019b. Word Usage Similarity Estimation with Sentence Representations and Automatic Substitutes. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 9–21, Minneapolis, Minnesota. Association for Computational Linguistics.

Aina Garí Soler, Anne Cocos, Marianna Apidianaki, and Chris Callison-Burch. 2019c. A Comparison of Context-sensitive Models for Lexical Substitution. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 271–282, Gothenburg, Sweden. Association for Computational Linguistics.

Bart Geurts. 2010. *Quantity implicatures*. Cambridge University Press.

Sahar Ghannay, Benoit Favre, Yannick Estève, and Nathalie Camelin. 2016. Word Embedding Evaluation and Combination. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 300–305, Portorož, Slovenia. European Language Resources Association (ELRA).

Mario Giulianelli, Marco Del Tredici, and Raquel Fernández. 2020. Analysing Lexical Semantic Change with Contextualised Word Representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3960–3973, Online. Association for Computational Linguistics.

Yoav Goldberg. 2019. Assessing BERT's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Nicole Gotzner, Stephanie Solt, and Anton Benz. 2018. Scalar diversity, negative strengthening, and adjectival semantics. *Frontiers in psychology*, 9:1659.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning Word Vectors for 157 Languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610.

Derek Gross and Katherine J Miller. 1990. Adjectives in WordNet. *International Journal of lexicography*, 3(4):265–277.

Emiliano Guevara. 2010. A Regression Model of Adjective-Noun Compositionality in Distributional Semantics. In *Proceedings of the 2010 Workshop on GEometrical Models of Natural Language Semantics*, pages 33–37, Uppsala, Sweden. Association for Computational Linguistics.

Abhijeet Gupta, Gemma Boleda, Marco Baroni, and Sebastian Padó. 2015. Distributional vectors encode referential attributes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 12–21, Lisbon, Portugal. Association for Computational Linguistics.

Guy Halawi, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren. 2012. Large-Scale Learning of Word Relatedness with Constraints. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1406–1414.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

John A Hartigan, Pamela M Hartigan, et al. 1985. The dip test of unimodality. *The annals of Statistics*, 13(1):70–84.

Mika Hasegawa, Tetsunori Kobayashi, and Yoshihiko Hayashi. 2020. Word Attribute Prediction Enhanced by Lexical Entailment Tasks. In *Proceedings of the 12th Language Resources and*

Evaluation Conference, pages 5846–5854, Marseille, France. European Language Resources Association.

Vasileios Hatzivassiloglou and Kathleen R. McKeown. 1993. Towards the Automatic Identification of Adjectival Scales: Clustering Adjectives According to Meaning. In *31st Annual Meeting of the Association for Computational Linguistics*, pages 172–182, Columbus, Ohio, USA. Association for Computational Linguistics.

Marti A. Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In *COLING 1992 Volume 2: The 15th International Conference on Computational Linguistics*.

Aurélie Herbelot and Eva Maria Vecchi. 2015. From concepts to models: some issues in quantifying feature norms. *Linguistic Issues in Language Technology (LiLT)*, 2(4).

John Hewitt and Percy Liang. 2019. Designing and Interpreting Probes with Control Tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A Structural Probe for Finding Syntax in Word Representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation. *Computational Linguistics*, 41(4):665–695.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Eric Huang, Richard Socher, Christopher Manning, and Andrew Ng. 2012. Improving Word Representations via Global Context and Multiple Word Prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 873–882, Jeju Island, Korea. Association for Computational Linguistics.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. SensEmbed: Learning Sense Embeddings for Word and Relational Similarity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 95–105, Beijing, China. Association for Computational Linguistics.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for Word Sense Disambiguation: An Evaluation Study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 897–907, Berlin, Germany. Association for Computational Linguistics.

Nancy Ide, Collin Baker, Christiane Fellbaum, Charles Fillmore, and Rebecca Passonneau. 2008. MASC: the Manually Annotated Sub-Corpus of American English. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Alexander Jakubowski, Milica Gasic, and Marcus Zibrowius. 2020. Topology of Word Embeddings: Singularities Reflect Polysemy. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 103–113, Barcelona, Spain (Online). Association for Computational Linguistics.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

David Jurgens. 2013. Embracing Ambiguity: A Comparison of Annotation Methodologies for Crowdsourcing Word Sense Labels. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 556–562, Atlanta, Georgia. Association for Computational Linguistics.

David Jurgens. 2014. An analysis of ambiguity in word sense annotations. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 3006–3012, Reykjavik, Iceland. European Language Resources Association (ELRA).

David Jurgens and Ioannis Klapaftis. 2013. Semeval-2013 task 13: Word sense induction for graded and non-graded senses. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, pages 290–299.

Andrej Karpathy, Justin Johnson, and Li Fei-Fei. 2015. Visualizing and Understanding Recurrent Networks. *ArXiv*, abs/1506.02078.

Jerrold J Katz and Jerry A Fodor. 1963. The structure of a semantic theory. *language*, 39(2):170–210.

Adam Kilgarriff. 1997. I don't believe in word senses. *Computers and the Humanities*, 31(2):91–113.

Adam Kilgarriff. 2004. How Dominant Is the Commonest Sense of a Word? Lecture Notes in Computer Science (vol. 3206), Text, Speech and Dialogue, Sojka Petr, Kopeček Ivan, Pala Karel (eds.), pages 103–112. Springer, Berlin, Heidelberg.

Joo-Kyung Kim and Marie-Catherine de Marneffe. 2013. Deriving Adjectival Scales from Continuous Space Word Representations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1625–1630, Seattle, Washington, USA. Association for Computational Linguistics.

Najoung Kim, Roma Patel, Adam Poliak, Patrick Xia, Alex Wang, Tom McCoy, Ian Tenney, Alexis Ross, Tal Linzen, Benjamin Van Durme, Samuel R. Bowman, and Ellie Pavlick. 2019. Probing What Different NLP Tasks Teach Machines about Function Word Comprehension. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 235–249, Minneapolis, Minnesota. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Walter Kintsch. 2001. Predication. *Cognitive science*, 25(2):173–202.

Kazuaki Kishida. 2005. *Property of average precision and its generalization: An examination of evaluation indicator for information retrieval experiments*. Technical Report NII-2005-014E, National Institute of Informatics Tokyo, Japan.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Arne Köhn. 2015. What's in an Embedding? Analyzing Word Embeddings through Multilingual Evaluation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2067–2073, Lisbon, Portugal. Association for Computational Linguistics.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(4):359–389.

John Koutsikakis, Ilias Chalkidis, Prodromos Malakasiotis, and Ion Androutsopoulos. 2020. GREEK-BERT: The Greeks visiting Sesame Street. In *Proceedings of the 11th Hellenic Conference on Artificial Intelligence (SETN 2020)*, pages 110–117, Athens, Greece.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the Dark Secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4365–4374, Hong Kong, China. Association for Computational Linguistics.

Charles W Kreidler. 1998. *Introducing english semantics*. Psychology Press.

Gerhard Kremer, Katrin Erk, Sebastian Padó, and Stefan Thater. 2014. What Substitutes Tell Us - Analysis of an "All-Words" Lexical Substitution Corpus. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 540–549, Gothenburg, Sweden. Association for Computational Linguistics.

Ramesh Krishnamurthy and Diane Nicholls. 2000. Peeling an Onion: The Lexicographer's Experience of Manual Sense-Tagging. *Computers and the Humanities*, 34(1-2):85–97.

George Lakoff and Mark Johnson. 2008. *Metaphors we live by*. University of Chicago press.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations.

Thomas K Landauer and Susan T Dumais. 1997. A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. *Psychological review*, 104(2):211.

Jey Han Lau and Timothy Baldwin. 2016. An empirical evaluation of doc2vec with practical insights into document embedding generation. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 78–86, Berlin, Germany. Association for Computational Linguistics.

Anne Lauscher, Goran Glavaš, Simone Paolo Ponzetto, and Ivan Vulić. 2020. A General Framework for Implicit and Explicit Debiasing of Distributional Word Vector Spaces. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, New York City, NY, USA.

Anne Lauscher, Ivan Vulić, Edoardo Maria Ponti, Anna Korhonen, and Goran Glavaš. 2019. Informing Unsupervised Pretraining with External Linguistic Knowledge. *arXiv preprint arXiv:1909.02339*.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. 2020. FlauBERT: Unsupervised Language Model Pre-training for French. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2479–2490, Marseille, France. European Language Resources Association.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International conference on Machine Learning*, pages 1188–1196, Beijing, China.

Claudia Leacock, Martin Chodorow, and George A. Miller. 1998. Using Corpus Statistics and WordNet Relations for Sense Identification. *Computational Linguistics*, 24(1):147–165.

Yoav Levine, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2020. SenseBERT: Driving Some Sense into BERT.

In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4656–4667, Online. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014a. Dependency-Based Word Embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014b. Linguistic Regularities in Sparse and Explicit Word Representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 171–180, Ann Arbor, Michigan. Association for Computational Linguistics.

Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. Do Supervised Distributional Methods Really Learn Lexical Inference Relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 970–976, Denver, Colorado. Association for Computational Linguistics.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and Understanding Neural Models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691, San Diego, California. Association for Computational Linguistics.

Jiwei Li and Dan Jurafsky. 2015. Do Multi-Sense Embeddings Improve Natural Language Understanding? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1722–1732, Lisbon, Portugal. Association for Computational Linguistics.

Xian Li, Paul Michel, Antonios Anastasopoulos, Yonatan Belinkov, Nadir Durrani, Orhan Firat, Philipp Koehn, Graham Neubig, Juan Pino, and Hassan Sajjad. 2019. Findings of the First Shared Task on Machine Translation Robustness. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 91–102, Florence, Italy. Association for Computational Linguistics.

Dekang Lin, Shaojun Zhao, Lijuan Qin, and Ming Zhou. 2003. Identifying Synonyms among Distributionally Similar Words. In *IJCAI*, volume 3, pages 1492–1493. Citeseer.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic Knowledge and Transferability of Contextual Representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2015. Learning context-sensitive word embeddings with neural tensor skip-gram model. In *IJCAI*, pages 1284–1290.

Qianchu Liu, Diana McCarthy, and Anna Korhonen. 2020. Towards Better Context-aware Lexical Semantics:Adjusting Contextualized Representations through Static Anchors. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4066–4075, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Daniel Loureiro and Jose Camacho-Collados. 2020. Don't Neglect the Obvious: On the Role of Unambiguous Words in Word Sense Disambiguation. *arXiv preprint:2004.14325*.

Daniel Loureiro, Kiamehr Rezaee, Mohammad Taher Pilehvar, and Jose Camacho-Collados. 2020. Language models and word sense disambiguation: An overview and analysis. *arXiv preprint arXiv:2008.11608*.

Li Lucy and Jon Gauthier. 2017. Are Distributional Representations Ready for the Real World? Evaluating Word Vectors for Grounded Perceptual Meaning. In *Proceedings of the First Workshop on Language Grounding for Robotics*, pages 76–85, Vancouver, Canada. Association for Computational Linguistics.

Marco Lui, Timothy Baldwin, and Diana McCarthy. 2012. Unsupervised estimation of word usage similarity. In *Proceedings of the Australasian Language Technology Association Workshop 2012*, pages 33–41, Dunedin, New Zealand.

Kevin Lund and Curt Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior research methods, instruments, & computers*, 28(2):203–208.

Minh-Thang Luong, Richard Socher, and Christopher D Manning. 2013. Better Word Representations with Recursive Neural Networks for Morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113.

John Lyons. 1995. *Linguistic semantics: An introduction*. Cambridge University Press.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Suresh Manandhar, Ioannis Klapaftis, Dmitriy Dligach, and Sameer Pradhan. 2010. Semeval-2010 task 14: Word sense induction &disambiguation. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 63–68.

Massimiliano Mancini, José Camacho-Collados, Ignacio Iacobacci, and Roberto Navigli. 2017. Embedding Words and Senses Together via Joint Knowledge-Enhanced Training. In *Proc. of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 100–111, Vancouver, Canada.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Marie-Catherine de Marneffe, Christopher D. Manning, and Christopher Potts. 2010. "Was It Good? It Was Provocative." Learning the Meaning of Scalar Adjectives". In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 167–176, Uppsala, Sweden. Association for Computational Linguistics.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a Tasty French Language Model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Matej Martinc, Syrielle Montariol, Elaine Zosa, and Lidia Pivovarova. 2020. Capturing Evolution in Word Usage: Just Add More Clusters? In *Companion Proceedings of the Web Conference 2020*, pages 343–349.

Héctor Martínez Alonso, Anders Johannsen, Oier Lopez de Lacalle, and Eneko Agirre. 2015. Predicting word sense annotation agreement. In *Proceedings of the First Workshop on Linking Computational Models of Lexical, Sentential and Discourse-level Semantics*, pages 89–94, Lisbon, Portugal. Association for Computational Linguistics.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Diana McCarthy. 2002. Lexical Substitution as a Task for WSD Evaluation. In *Proceedings of the ACL-02 Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, pages 089–115. Association for Computational Linguistics.

Diana McCarthy, Marianna Apidianaki, and Katrin Erk. 2016. Word Sense Clustering and Clusterability. *Computational Linguistics*, 42(2):245–275.

Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2004. Finding Predominant Word Senses in Untagged Text. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 279–286, Barcelona, Spain.

Diana McCarthy and Roberto Navigli. 2007. SemEval-2007 Task 10: English Lexical Substitution Task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 48–53, Prague, Czech Republic. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In *Proceedings of the 57th Annual*

Meeting of the Association for Computational Linguistics, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

John P McCrae, Ian Wood, and Amanda Hicks. 2017. The Colloquial WordNet: Extending Princeton WordNet with Neologisms. In *International Conference on Language, Data and Knowledge*, pages 194–202. Springer.

Louise McNally. 2016. Scalar alternatives and scalar inference involving adjectives: A comment on van Tiel, et al. 2016. In Ruth Kramer Jason Ostrove and Joseph Sabbagh, editors, *Asking the Right Questions: Essays in Honor of Sandra Chung*, pages 17–28.

Louise McNally and Gemma Boleda. 2004. Relational adjectives as properties of kinds. Colloque de Syntaxe et Sémantique à Paris.

Ken McRae, George Cree, Mark Seidenberg, and Chris Mcnorgan. 2005. Semantic feature production norms for a large set of living and nonliving things. *Behavior research methods*, 37:547–59.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning Generic Context Embedding with Bidirectional LSTM. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61, Berlin, Germany. Association for Computational Linguistics.

Oren Melamud, Omer Levy, and Ido Dagan. 2015. A Simple Word Embedding Model for Lexical Substitution. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 1–7, Denver, Colorado. Association for Computational Linguistics.

Gerard de Melo and Mohit Bansal. 2013. Good, Great, Excellent: Global Inference of Semantic Intensities. *Transactions of the Association for Computational Linguistics*, 1:279–290.

Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K Gray, Joseph P Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, et al. 2011. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Timothee Mickus, Denis Paperno, Mathieu Constant, and Kees van Deemter. 2020. What do you mean, BERT? In *Proceedings of the Society for Computation in Linguistics 2020*, pages 279–290, New York, New York. Association for Computational Linguistics.

Rada Mihalcea, Ravi Sinha, and Diana McCarthy. 2010. SemEval-2010 Task 2: Cross-Lingual Lexical Substitution. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 9–14, Uppsala, Sweden. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint:1301.3781v3*.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in Pre-Training Distributed Word Representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, Lake Tahoe, Nevada.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41.

George A Miller and Walter G Charles. 1991. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A Semantic Concordance. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244, Columbus, Ohio. Association for Computational Linguistics.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cogn. Sci.*, 34(8):1388–1429.

Saif Mohammad. 2018. Word Affect Intensities. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

G Craig Murray and Rebecca Green. 2004. Lexical knowledge and human disagreement on a WSD task. *Computer Speech & Language*, 18(3):209–222.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 95–100, Montréal, Canada. Association for Computational Linguistics.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. *Artificial Intelligence*, 193:217–250.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1059–1069, Doha, Qatar. Association for Computational Linguistics.

Finn Årup Nielsen. 2011. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. In *Proceedings of the ESWC 2011 Workshop on 'Making Sense of Microposts: Big things come in small packages'*, volume 718 in CEUR Workshop Proceedings, pages 93–98.

Rafail Ostrovsky, Yuval Rabani, Leonard J Schulman, and Chaitanya Swamy. 2012. The effectiveness of Lloyd-type methods for the k-means problem. *Journal of the ACM (JACM)*, 59(6):28.

Sebastian Padó and Mirella Lapata. 2007. Dependency-Based Construction of Semantic Space Models. *Computational Linguistics*, 33(2):161–199.

Martha Palmer, Olga Babko-Malaya, and Hoa Trang Dang. 2004. Different Sense Granularities for Different Applications. In *Proceedings of the 2nd International Workshop on Scalable Natural Language Understanding (ScaNaLU 2004) at HLT-NAACL 2004*, pages 49–56, Boston, Massachusetts, USA. Association for Computational Linguistics.

Martha Palmer, Hoa Trang Dang, and Christiane Fellbaum. 2007. Making fine-grained and coarse-grained sense distinctions, both manually and automatically. *Journal of Natural Language Engineering*, 13(2):137–163.

Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1–2):1–135.

Patrick Pantel and Dekang Lin. 2002. Discovering word senses from text. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 613–619.

Rebecca Passonneau, Ansaf Salleb-Aouissi, and Nancy Ide. 2009. Making Sense of Word Sense Variation. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, pages 2–9, Boulder, Colorado. Association for Computational Linguistics.

Rebecca J Passonneau, Vikas Bhardwaj, Ansaf Salleb-Aouissi, and Nancy Ide. 2012. Multiplicity and Word Sense: Evaluating and Learning from Multiply Labeled Word Sense Annotations. *Language Resources and Evaluation*, 46(2):219–252.

Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. *arXiv preprint arXiv:1404.5367*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas

Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Ajay Patel, Alexander Sands, Chris Callison-Burch, and Marianna Apidianaki. 2018. Magnitude: A Fast, Efficient Universal Vector Embedding Utility Package. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 120–126, Brussels, Belgium. Association for Computational Linguistics.

Ellie Pavlick and Chris Callison-Burch. 2016. Most "babies" are "little" and most "problems" are "huge": Compositional Entailment in Adjective-Nouns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2164–2173, Berlin, Germany. Association for Computational Linguistics.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification . In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430, Beijing, China. Association for Computational Linguistics.

Maria Pelevina, Nikolay Arefiev, Chris Biemann, and Alexander Panchenko. 2016. Making Sense of Word Embeddings. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 174–183, Berlin, Germany. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018b. Dissecting Contextual Word Embeddings: Architecture and Representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Brussels, Belgium. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019a. Knowledge Enhanced Contextual Word Representations. In

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 43–54, Hong Kong, China. Association for Computational Linguistics.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019b. To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language Models as Knowledge Bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Mohammad Taher Pilehvar and José Camacho-Collados. 2018. Wic: 10, 000 example pairs for evaluating context-sensitive representations. *CoRR*, abs/1808.09121.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the Word-in-Context Dataset for Evaluating Context-Sensitive Meaning Representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

Mohammad Taher Pilehvar and Nigel Collier. 2016. De-conflated semantic representations. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1680–1690, Austin, Texas. Association for Computational Linguistics.

Tiago Pimentel, Rowan Hall Maudslay, Damian Blasi, and Ryan Cotterell. 2020. Speakers Fill Lexical Semantic Gaps with Context. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4004–4015, Online. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. *arXiv preprint arXiv:2003.07082*.

Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and Why Are Pre-Trained Word Embeddings Useful for Neural Machine Translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 529–535, New Orleans, Louisiana. Association for Computational Linguistics.

Chen Qu, Liu Yang, Minghui Qiu, W Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019. BERT with History Answer Embedding for Conversational Question Answering. In *Proceedings of*

the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 1133–1136.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI Blog*, 1(8):9.

Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. 2011. A word at a time: Computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on World wide web*, pages 337–346.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Alessandro Raganato, Tommaso Pasini, Jose Camacho-Collados, and Mohammad Taher Pilehvar. 2020. XL-WiC: A Multilingual Benchmark for Evaluating Semantic Contextualization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7193–7206.

Alessandro Raganato and Jörg Tiedemann. 2018. An Analysis of Encoder Representations in Transformer-Based Machine Translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297, Brussels, Belgium. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null It Out: Guarding Protected Attributes by Iterative Nullspace Projection. *arXiv preprint arXiv:2004.07667*.

Abhilasha Ravichander, Eduard Hovy, Kaheer Suleman, Adam Trischler, and Jackie Chi Kit Cheung. 2020. On the Systematicity of Probing Contextualized Word Representations: The Case of Hypernymy in BERT. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 88–102, Barcelona, Spain (Online). Association for Computational Linguistics.

Siva Reddy, Ioannis Klapaftis, Diana McCarthy, and Suresh Manandhar. 2011a. Dynamic and Static Prototype Vectors for Semantic Composition. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 705–713, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Siva Reddy, Diana McCarthy, Suresh Manandhar, and Spandana Gella. 2011b. Exemplar-Based Word-Space Model for Compositionality Detection: Shared Task System Description. In *Proceedings of the Workshop on Distributional Semantics and Compositionality*, pages 54–60, Portland, Oregon, USA. Association for Computational Linguistics.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and Measuring the Geometry of BERT. In *Advances in Neural Information Processing Systems*, pages 8592–8600, Vancouver, Canada.

Joseph Reisinger and Raymond J. Mooney. 2010. Multi-Prototype Vector-Space Models of Word Meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117, Los Angeles, California. Association for Computational Linguistics.

Philip Resnik and David Yarowsky. 1999. Distinguishing systems and distinguishing senses: New evaluation methods for word sense disambiguation. *Natural language engineering*, 5(2):113–133.

Sven Rill, J. vom Scheidt, Johannes Drescher, Oliver Schütz, Dirk Reinel, and Florian Wogenstein. 2012. A generic approach to generate opinion lists of phrases for opinion mining applications. In *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining (WISDOM)*, pages 1–8, Beijing, China.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Stephen Roller and Katrin Erk. 2016. Relations such as Hypernymy: Identifying and Exploiting Hearst Patterns in Distributional Vectors for Lexical Entailment. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2163–2172, Austin, Texas. Association for Computational Linguistics.

Eleanor Rosch. 1975. Cognitive Representations of Semantic Categories. *Journal of experimental psychology: General*, 104(3):192.

Alex Rosenfeld and Katrin Erk. 2018. Deep Neural Models of Semantic Shift. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 474–484, New Orleans, Louisiana. Association for Computational Linguistics.

Sascha Rothe and Hinrich Schütze. 2015. AutoExtend: Extending Word Embeddings to Embeddings for Synsets and Lexemes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1793–1803, Beijing, China. Association for Computational Linguistics.

Peter J Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Herbert Rubenstein and John B Goodenough. 1965. Contextual Correlates of Synonymy. *Communications of the ACM*, 8(10):627–633.

Dana Rubinstein, Effi Levi, Roy Schwartz, and Ari Rappoport. 2015. How Well Do Distributional Models Capture Different Types of Semantic Knowledge? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 726–730, Beijing, China. Association for Computational Linguistics.

Josef Ruppenhofer, Michael Wiegand, and Jasper Brandes. 2014. Comparing methods for deriving intensity scores for adjectives. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 117–122, Gothenburg, Sweden. Association for Computational Linguistics.

Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18(11):613–620.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.

Timo Schick and Hinrich Schütze. 2020. Rare Words: A Major Problem for Contextualized Embeddings and How to Fix it by Attentive Mimicking. In *AAAI*, pages 8766–8774.

Dominik Schlechtweg, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. 2020. SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1–23, Barcelona (online). International Committee for Computational Linguistics.

Karin Kipper Schuler. 2006. *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. Ph.D. thesis, University of Pennsylvania.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE.

Hinrich Schütze. 1998. Automatic Word Sense Discrimination. *Computational linguistics*, 24(1):97–123.

Hinrich Schütze and Jan Pedersen. 1993. A vector model for syntagmatic and paradigmatic relatedness. In *Proceedings of the 9th Annual Conference of the UW Centre for the New OED and Text Research*, pages 104–113. Citeseer.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association*

for Computational Linguistics (Volume 1: Long Papers), pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Raksha Sharma, Mohit Gupta, Astha Agarwal, and Pushpak Bhattacharyya. 2015. Adjective Intensity and Sentiment Analysis. In *Proceedings of the 2015 Conference on Empirical Methods for Natural Language Processing*, pages 2520–2526, Lisbon, Portugal. Association for Computational Linguistics.

Serge Sharoff. 2006. Open-source corpora: Using the net to fish for linguistic data. *International journal of corpus linguistics*, 11(4):435–462.

Vera Sheinman, Christiane Fellbaum, Isaac Julien, Peter Schulam, and Takenobu Tokunaga. 2013. Large, huge or gigantic? Identifying and encoding intensity relations among adjectives in WordNet. *Language resources and evaluation*, 47(3):797–816.

Vera Sheinman and Takenobu Tokunaga. 2009. AdjScales: Visualizing Differences between Adjectives for Language Learners. *IEICE Transactions on Information and Systems*, 92-D:1542–1550.

Weijia Shi, Muhao Chen, Pei Zhou, and Kai-Wei Chang. 2019. Retrofitting Contextualized Word Embeddings with Paraphrases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1198–1203, Hong Kong, China. Association for Computational Linguistics.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does String-Based Neural MT Learn Source Syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, Austin, Texas. Association for Computational Linguistics.

Chaitanya Shivade, Marie-Catherine de Marneffe, Eric Fosler-Lussier, and Albert M. Lai. 2015. Corpus-based discovery of semantic intensity scales. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–493, Denver, Colorado. Association for Computational Linguistics.

Ravi Sinha and Rada Mihalcea. 2014. Explorations in lexical sample and all-words lexical substitution. *Natural Language Engineering*, 20(1):99.

Barry Smith and Christiane Fellbaum. 2004. Medical WordNet: A New Methodology for the Construction and Validation of Information Resources for Consumer Health. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 371–382, Geneva, Switzerland. COLING.

Samuel L Smith, David HP Turban, Steven Hamblin, and Nils Y Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax.

Robyn Speer and Catherine Havasi. 2012. Representing General Relational Knowledge in ConceptNet 5. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3679–3686, Istanbul, Turkey. European Language Resources Association (ELRA).

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. In *Proceedings of the 7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*, Cardiff, UK. Leibniz-Institut für Deutsche Sprache.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-Based Methods for Sentiment Analysis. *Computational linguistics*, 37(2):267–307.

Kaveh Taghipour and Hwee Tou Ng. 2015. Semi-Supervised Word Sense Disambiguation Using Word Embeddings in General and Specific Domains. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 314–323, Denver, Colorado. Association for Computational Linguistics.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. oLMpics-On What Language Model Pre-training Captures. *Transactions of the Association for Computational Linguistics*, 8:743–758.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Wilson L Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. BERT Rediscovers the Classical NLP Pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Stefan Thater, Georgiana Dinu, and Manfred Pinkal. 2009. Ranking Paraphrases in Context. In *Proceedings of the 2009 Workshop on Applied Textual Inference (TextInfer)*, pages 44–47, Suntec, Singapore. Association for Computational Linguistics.

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. 2010. Contextualizing Semantic Representations Using Syntactically Enriched Vector Models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 948–957, Uppsala, Sweden. Association for Computational Linguistics.

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. 2011. Word Meaning in Context: A Simple and Effective Vector Model. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1134–1143, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Fei Tian, Hanjun Dai, Jiang Bian, Bin Gao, Rui Zhang, Enhong Chen, and Tie-Yan Liu. 2014. A Probabilistic Model for Learning Multi-Prototype Word Embeddings. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 151–160, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

David Tuggy. 1993. Ambiguity, polysemy, and vagueness. *Cognitive Linguistics (includes Cognitive Linguistic Bibliography)*, 4(3):273–290.

Peter Turney. 2008. A Uniform Approach to Analogies, Synonyms, Antonyms, and Associations. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 905–912, Manchester, UK. Coling 2008 Organizing Committee.

Peter D Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188.

Bob Van Tiel, Emiel Van Miltenburg, Natalia Zevakhina, and Bart Geurts. 2016. Scalar Diversity. *Journal of semantics*, 33(1):137–175.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, Long Beach, California, USA.

Jean Véronis. 1998. A study of polysemy judgements and inter-annotator agreement. In *Programme and advanced papers of the Senseval workshop*, pages 2–4. Citeseer.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: BERT for Finnish. *arXiv preprint arXiv:1912.07076*.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019a. The Bottom-up Evolution of Representations in the Transformer: A Study with Machine Translation and Language Modeling Objectives.

In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4396–4406, Hong Kong, China. Association for Computational Linguistics.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019b. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Elena Voita and Ivan Titov. 2020. Information-Theoretic Probing with Minimum Description Length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online. Association for Computational Linguistics.

Piek Vossen. 1998. EuroWordNet: A Multilingual Database with Lexical Semantic Networks. *Dordrecht: Kluwer Academic Publishers. doi*, 10:978–94.

Ivan Vulić, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. 2017. HyperLex: A Large-Scale Evaluation of Graded Lexical Entailment. *Computational Linguistics*, 43(4):781–835.

Ivan Vulić and Nikola Mrkšić. 2018. Specialising Word Vectors for Lexical Entailment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1134–1145, New Orleans, Louisiana. Association for Computational Linguistics.

Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. Probing Pretrained Language Models for Lexical Semantics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7222–7240, Online. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Yile Wang, Leyang Cui, and Yue Zhang. 2019b. How Can BERT Help Lexical Semantics Tasks? *arXiv preprint arXiv:1911.02929*.

Gregor Wiedemann, Steffen Remus, Avi Chawla, and Chris Biemann. 2019. Does BERT Make Any Sense? Interpretable Word Sense Disambiguation with Contextualized Embeddings. In

Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019): Long Papers, pages 161–170, Erlangen, Germany. German Society for Computational Linguistics & Language Technology.

Anna Wierzbicka. 1972. Semantic primitives.

Bryan Wilkinson. 2017. *Identifying and Ordering Scalar Adjectives Using Lexical Substitution*. Ph.D. thesis, University of Maryland, Baltimore County.

Bryan Wilkinson and Tim Oates. 2016. A Gold Standard for Scalar Adjectives. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2669–2675, Portorož, Slovenia. European Language Resources Association (ELRA).

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv preprint:1609.08144*.

Christos Xypolopoulos, Antoine Tixier, and Michalis Vazirgiannis. 2021. Unsupervised Word Polysemy Quantification with Multiresolution Grids of Contextual Embeddings. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3391–3401, Online. Association for Computational Linguistics.

Yiben Yang, Larry Birnbaum, Ji-Ping Wang, and Doug Downey. 2018. Extracting Commonsense Properties from Embeddings with Limited Human Guidance. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 644–649, Melbourne, Australia. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

David Yenicelik, Florian Schmidt, and Yannic Kilcher. 2020. How does BERT capture semantics? A closer look at polysemous words. In *Proceedings of the Third BlackboxNLP Workshop*

on Analyzing and Interpreting Neural Networks for NLP, pages 156–162, Online. Association for Computational Linguistics.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Bin Zhang. 2001. Dependence of Clustering Algorithm Performance on Clustered-ness of Data. *HP Labs Technical Report HPL-2001-91*.

Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. 2018. Learning Gender-Neutral Word Embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4847–4853, Brussels, Belgium. Association for Computational Linguistics.

Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. 2019. BERT-based Lexical Substitution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3368–3373, Florence, Italy. Association for Computational Linguistics.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV'15)*, page 19–27, Santiago, Chile. IEEE Computer Society.

George Kingsley Zipf. 1945. The meaning-frequency relationship of words. *Journal of General Psychology*, 33(2):251–256.

Will Y. Zou, Richard Socher, Daniel Cer, and Christopher D. Manning. 2013. Bilingual Word Embeddings for Phrase-Based Machine Translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1393–1398, Seattle, Washington, USA. Association for Computational Linguistics.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Représentation du Sens des Mots dans les Modèles de Langue Neuronaux : Polysémie Lexicale et Relations Sémantiques

Mots clés: sémantique lexicale, traitement automatique des langues, modèles de langue, plongements lexicaux, représentations contextualisées, polysémie

Résumé: Les plongements contextualisés représentent l'usage des mots dans leur contexte. Nous étudions les connaissances liées au sens des mots encodées dans ces représentations et proposons des méthodes pour améliorer leur qualité. Nous nous appuyons sur des expériences qui traitent de la similarité des usages des mots et des annotations contenant des substituts lexicaux attribuées par les modèles à des usages des mots en contexte. Nous évaluons les représentations sur les tâches de prédiction de la similarité des usages des mots, de la possibilité de regroupement de leur sens, et de leur niveau de polysémie. Nous explorons aussi des relations sémantiques : la relation d'intensité entre adjectifs scalaires et les propriétés de concepts nominaux, exprimées par leur modificateurs adjectivaux. Nous ménons des expériences avec des modèles multilingues et monolingues dans différentes langues et des plongements statiques. Nous montrons que les représentations contextualisées encodent des connaissances riches sur le sens des mots et leur relations sémantiques acquises lors de l'entraînement, qui sont enrichies par des informations provenant de nouveaux contextes.

Title: Word Meaning Representation in Neural Language Models: Lexical Polysemy and Semantic Relationships

Keywords: lexical semantics, natural language processing, language models, word embeddings, contextualised representations, polysemy

Abstract: Contextual language models generate representations for word instances. We investigate the knowledge about word meaning encoded in these representations and propose methods to automatically enhance their quality with external semantic knowledge. We access the polysemy information in contextualised representations through usage similarity experiments and automatic substitute annotations assigned by the models to words in context. We evaluate their quality on the tasks of usage similarity, word sense clusterability and polysemy level prediction. Furthermore, we explore semantic relationships. We specifically address scalar adjective intensity and noun properties as expressed in their adjectival modifiers. Our experiments involve multilingual and multilingual contextual language models in different languages, and static embeddings. We show that contextualised representations encode rich knowledge about word meaning and semantic relationships acquired during training and enriched with information from new contexts of use.