Détection de topiques et leur similarité dans les textes scientifiques

Simon David Hernandez Perez

UNIVERSITÉ PARIS 13

UNIVERSITÉ PARIS 13

Institut Galilée

Laboratoire d'Informatique de Paris Nord

DOCTORAT
Discipline: Informatique

Thèse présentée par **Simon David HERNANDEZ PEREZ**
Soutenue le **20 décembre 2019**

Détection de topiques et leur similarité dans les textes scientifiques

Topic detection and similarity from scientific literature

Rapporteurs	Pr. Delphine BATTISTELLI
	Dr. Nathalie AUSSENAC-GILLES
Examinateurs	Pr. Benoît CRABBE
	Pr. Nathalie PERNELLE
Directeur de thèse	Pr. Thierry CHARNOIS
Co-directeur de thèse	MCF. Davide BUSCALDI

Abstract

Nowadays, it is increasingly difficult for researchers to find the state of the art of their respective fields of study, mostly due to the amount of scientific documents generated in the world every day. It is challenging and highly time-consuming to curate and index scientific literature, mostly because it is required wide knowledge and expertness. Currently, there are services like ScienceDirect, Microsoft Academic Graph, Mendeley, Google Scholar, SpringerLink, etc., providing interfaces to browse along a vast collections of scientific publications facilitating and suggesting articles of interest to their users. Those services rely mostly on the lexical content of the documents and their metadata, like keywords, relationships between references, citations and authors. Using that information is effective when the topic to search is widely known and conventional concepts are addressed. Considering, researchers' work demands to push the boundaries of their fields of study, problems emerge when they need to find information about unconventional concepts, a situation that is not strange. Under this circumstance, there are distinct phenomena affecting the results in the semantic level, i.e., polysemy and synonymy, therefore, it is needed to measure semantic similarity on the content of the documents.

In the interest of easing the measuring of semantic similarity between scientific documents, there are recent works addressing the task of automatic keyphrase extraction (ACL RD-TEC 2.0, SemEval 2017 Task 10), using supervised and unsupervised approaches, mostly, based on heuristics (like ranking methods, rules, regular expressions), probabilistic approaches (like CRFs), classification or clusterization, and neural networks (like LSTM), being the latter the ones providing best results. To measure semantic similarity between terms and documents, there are statistical approaches (like LSA, PMI, LDA), Word Embeddings (like Word2Vec, FastText, GloVe) in combination with ontological resources (like WordNet, ConceptNet numberbatch). In addition, given that the access to scientific literature is usually restricted, there are notables efforts to concentrate public experimental data (ArnetMiner).

In this context, our first contribution is the experimental results of using part-of-speech tag sequences to filter candidate keyphrases in scientific documents. We improved satisfactorily the performance of CRFs trained using those filtered candidates. We extracted a set of part-of-speech tag sequences to filter candidates from scientific texts. Additionally, we implemented our approach in an open-source software package publicly available. We analized correlations of document similarity measures and find that measuring similarity centroids of word embeddings behave similarly using words and keyphrases. We also generated a subset of scientific abstracts from Arnet-Miner containing concepts (keyphrases or terms) with different lexical representations. Those concepts are extracted keyphrases, using our approach and package, matching terms from Wikipedia redirections.

Résumé

Pour un chercheur la recherche de documents scientifiques relatifs à l'état de l'art de son domaine est une tâche difficile, notamment en raison de la grande quantité de données publiées qui augmente chaque jour. D'un autre côté l'indexation et la structuration de tels documents sont des tâches couteuses en temps et requièrent une grande expertise et des connaissances des domaines. Actuellement, des services tels que ScienceDirect, Microsoft Academic Graph, Mendeley, Google Scholar, SpringerLink, etc., fournissent des interfaces pour parcourir une vaste collection de publications scientifiques permettant de proposer des articles intéressants pour leurs utilisateurs. Ces services reposent principalement sur le contenu lexical des documents et leurs métadonnées, comme les mots-clés, les relations entre références, les citations et auteurs. L'utilisation de ces informations est efficace lorsque le sujet de recherche est bien connu et des concepts conventionnels abordés. Cependant, pour repousser les limites de son champ de recherche, le chercheur fait face à des difficultés lorsqu'il recherche des informations sur des sujets ou concepts non conventionnels. En effet, les systèmes doivent pouvoir traiter des phénomènes linguistiques sémantiques tels que la polysémie et la synonymie, ce qui nécessite de pouvoir mesurer la similarité sémantique entre termes et entre documents.

Pour améliorer la mesure de similarité sémantique entre documents scientifiques, il existe des travaux récents portant sur la tâche d'extraction automatique d'expressions-clés (ACL RD-TEC 2.0, SemEval 2017 tâche 10). Ces travaux utilisent des approches supervisées et non supervisées, principalement basées sur des méthodes heuristiques (comme les méthodes de ranking, les règles, les expressions régulières), les approches probabilistes (telles que les CRF), les approches de classification ou de clustering, et les réseaux neuronaux (tels que LSTM), ces derniers offrant les meilleurs résultats. Pour calculer la mesure de similarité sémantique entre termes et documents, des approches statistiques (telles que LSA, PMI, LDA), Word Embeddings (notamment Word2Vec, FastText, GloVe) sont souvent combinées avec des ressources ontologiques (telles que WordNet, ConceptNet

numberbatch). De plus, des efforts notables ont été déployés pour fournir des données expérimentales publiques (ArnetMiner).

Dans ce contexte, notre première contribution concerne les résultats expérimentaux de l'utilisation de séquences d'étiquettes de catégorie grammaticale pour filtrer des candidates d'expressions-clés dans des documents scientifiques. Nous avons amélioré de manière satisfaisante les performances du CRF entraîné avec ces candidats filtrés. Nous avons ainsi extrait un ensemble de séquences d'étiquettes de catégorie grammaticale pour filtrer des candidats à partir de textes scientifiques. De plus, nous avons implémenté notre approche dans un logiciel libre accessible au public. Nous avons analysé les corrélations des mesures de similitude des documents et constaté que la mesure des centroïdes de similarité des Word Embeddings se comporte de la même manière en utilisant des mots et des expressions-clés. Nous avons également généré un sous-ensemble de résumés scientifiques à partir d'ArnetMiner contenant des concepts (expressions-clés ou termes) avec différentes représentations lexicales. Ces concepts sont des expressions-clés, extraites en utilisant notre approche et notre outil, et correspondent aux termes des redirections Wikipédia.

Contents

List of Figures

List of Tables

Glossary

automatic keyphrase extraction	Identification of keyphrases in a given text.. 22
keyphrase	Term related to the main topic of a text.. xv, 17, 18
keyphrase assignment	Assignment of keyphrases not present in the text.. 21
keyphrase extraction	Identification of keyphrases in a given text.. 20, 21
vocabulary	Set of possible words.. xv

Nomenclature

Description of symbols used within the body of the document.

Term extraction

K	Set of keyphrases.
k	Keyphrase.
V	Vocabulary.

Semantic similarity

D	Set of documents.

Other Symbols

ρ	Pearson correlation.

Part I

General Introduction

Chapter 1

Introduction

A *state of the art* from the perspective of researchers, refers to the scientific and engineering knowledge about a specific field of study in a given time. It is stated in the form of a set of scientific publications describing the methodologies and techniques used to define, analyze or solve the object of study.

Researchers often make use of services to browse among a vast number of scientific publications to search for the *state of the art* on their fields. They look for documents, querying specific terms or look for suggestions based on the documents that they already know. To provide those services, it is needed to *index* and *categorize* documents using keywords (multi-word terms) provided by the authors, the topics issued by the journals, the citation metadata and the raw content. Although these services are helpful, there is still room for improvement, in particular at the semantic level.

Despite the fact that *keywords* provided by the authors of a scientific document bring a general semantic description of it, they are not always useful to characterize a document against its similars when specificity is required. The causes vary, for example, it might be a consequence of the *generality* of the keywords or a poor criterion of the selection of the keywords. Hence, current approaches extract *keywords* (or *keyphrases*) within the body of a document instead of using those provided by the authors or the editors of a scientific publication. The automatic extraction of keyphrases from scientific publications is not an easy task and has its challenges. Considering that, the *keyphrases* of a document are the main terms representing a document (by definition), they should be more effective to characterize semantically a document.

Our intention is to use semantics to ease the search of the *state of the art* in scientific literature. Therefore, this work is focused on two tasks. First, extracting keyphrases (or keywords or multi-word terms) from scientific liter-

ature that "better" characterize the topics in the documents among scientific literature. Second, measuring the impact of semantic similarity measures on the ability to retrieve documents pertinent to a given topic, specified as a set of keyphrases.

In this chapter we describe the motivation of this project, see Section 1.1. In Sections 1.2 and 1.3 we make an explanation of the challenges and goals. Our contributions are described in Section 1.4. Additionally, in Section 1.5 there is a description of the content of the following chapters in this document.

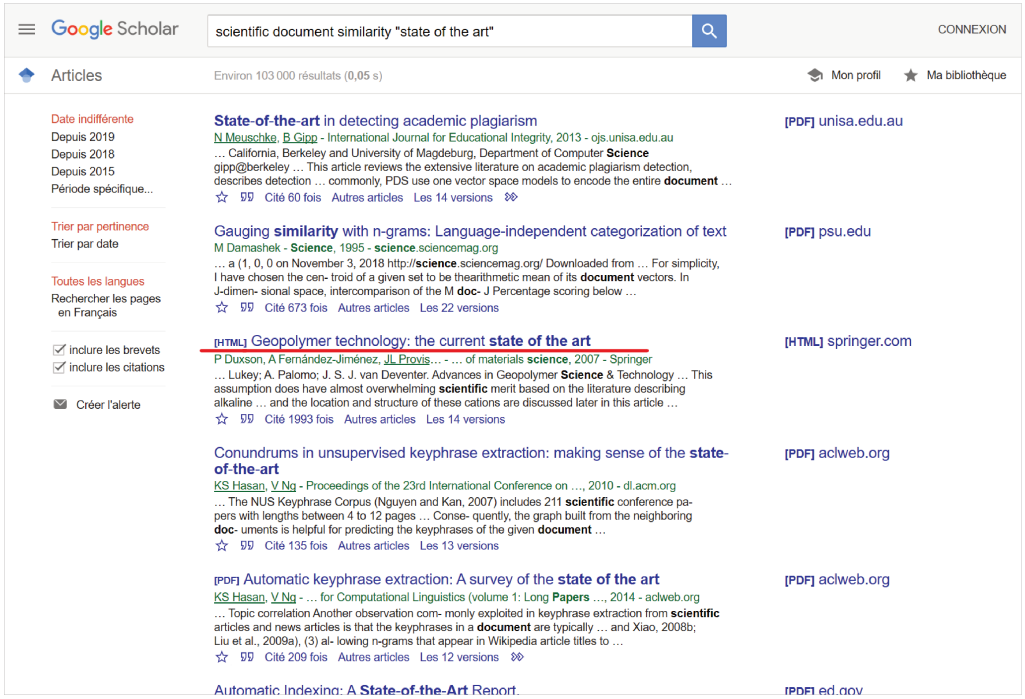
1.1 Motivation

Nowadays, the number of scientific publications is continuously growing, in all disciplines. According to Bjork et al. (2009), 1.35 million articles were published in indexed journals in the single year 2006, and the growth rate in the number of scientific publications has been estimated by Larsen and von Ins (2010) to be between 2.2% and 9% for journals and between 1.6% and 14% for conferences (depending on the disciplines) in the decade 1997-2007. It is becoming more and more difficult to search information required to write scientific papers, to review the work of other researchers, or to look for experts. Usually this kind of search involves checking the originality of an idea or a method. Current search engines dedicated to the exploration of scientific literature, such as Google scholar[1], Scopus[2], Microsoft Academic[3], Springer[4] and ArnetMiner[5] are based on text content, author and citation graphs.

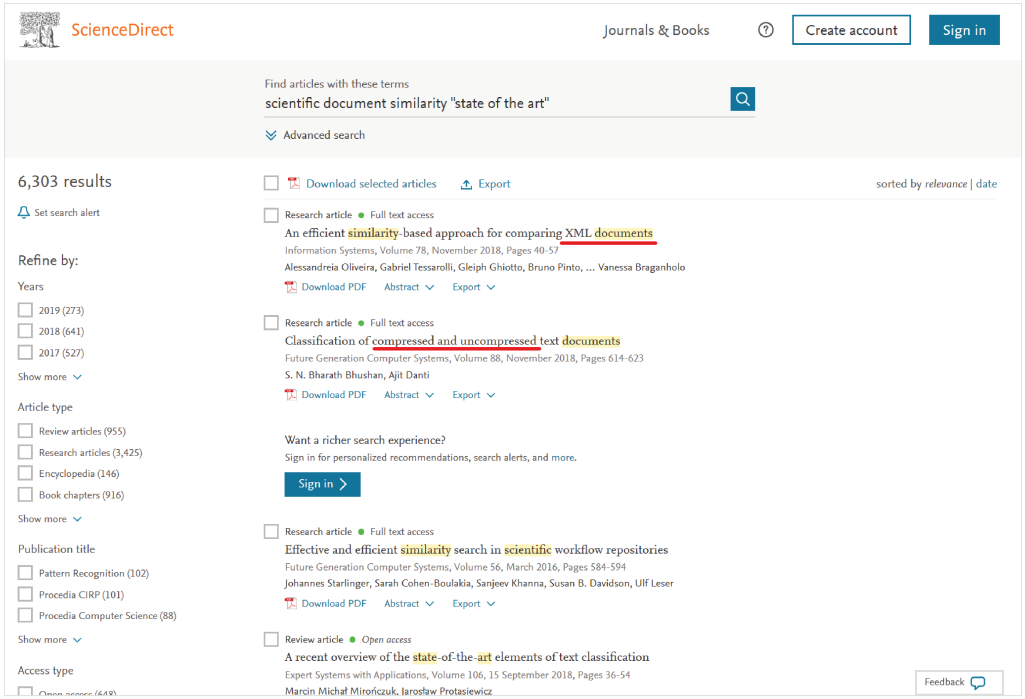
Despite those services and efforts, at this moment there is not an infallible service able to solve all the possible conditions affecting the search of information. (See example in Figures 1.1a and 1.1b.) A huge amount of work and time is still required, not only due to the number of available articles, but also because of the intellectual effort required by the task. Researchers look among texts that need deep understanding in very specialized fields of study, very often not fully related to their interests, because it is not always easy to discern if something is useful or not.

Recent efforts looking to enhance the access to scientific literature make use of techniques from the semantic web (Osborne and Motta 2015) and com-

[1] https://scholar.google.com
[2] https://www.scopus.com
[3] https://academic.microsoft.com
[4] https://www.springer.com
[5] https://aminer.org/

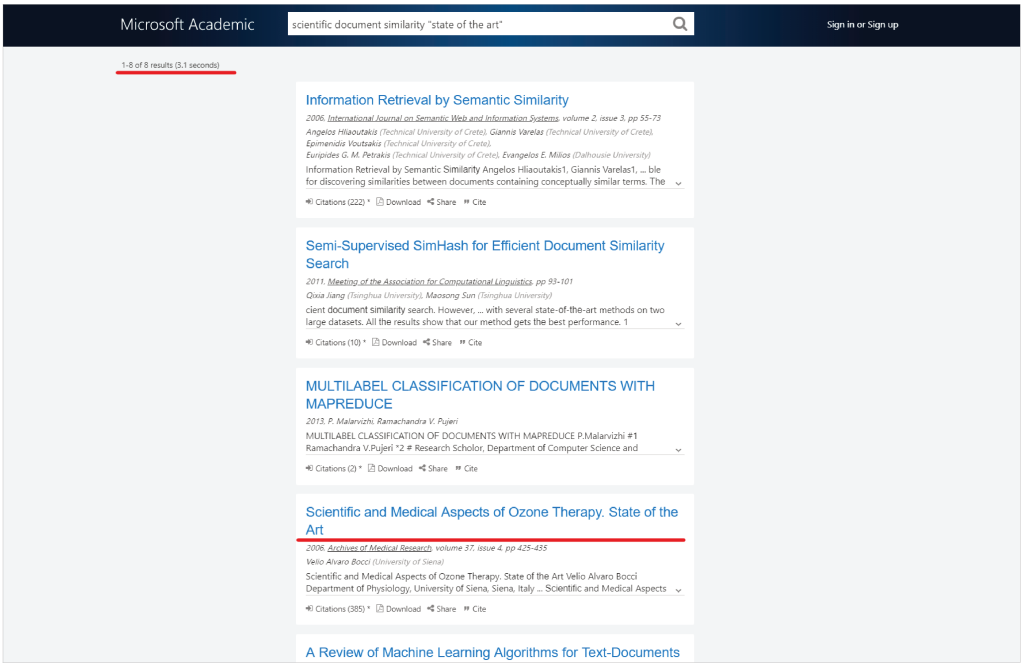


(a) *Google Scholar*

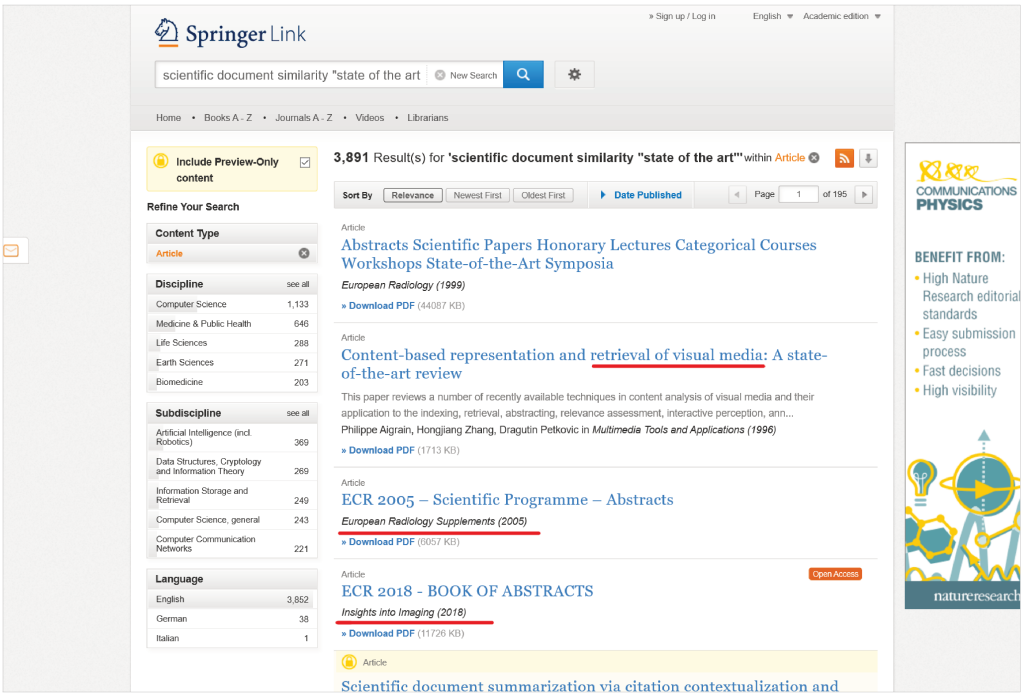


(b) *ScienceDirect*

Figure 1.1a *Examples of searches of the* state of the art *in widely used services.*



(c) *MicrosoftAcademic*



(d) *SpringerLink*

Figure 1.1b *Examples of searches of the* state of the art *in widely used services.*

binations of scientometry and natural language processing (Wolfram 2016). Some initiatives have been started to gather researchers around this problem, like the SAVE-SD[6] workshops and the ScienceIE task (Augenstein et al. 2017) at SemEval2017 Task 10[7].

To ease the search of the *state of the art*, it is needed to semantically compare documents. With this objective in mind, different approaches from natural language processing and information retrieval have been applied; however, as far as we know there is not a clear way to compare the effectiveness of those methods across domains, because forming an annotated corpus to measure their performance requires a lot of time and effort, as well as highly specialized annotators. Given that keyphrases have been largely considered a *"high-level description of a document's content"* (Frank et al. 1999), we consider that they can be used to measure semantically similarity. Their use has the advantage of dimensionality reduction, although it comes at the price of losing context information. Moreover, we do not know how the loss of context could affect the measuring of document semantic similarity in comparison with using the full content of a document. Consequently, we need a reliable method to extract keyphrases and a corpus of scientific documents to evaluate the performance of different measurements of semantic similarity.

1.2 Keyphrase extraction

The first part of our work consists in the extraction of keyphrases from scientific documents. It is usual that academic journals demand authors for a list of keyphrases for their articles (Frank et al. 1999; Peter D Turney 2000), to facilitate the categorization and search of the document. Keyphrases are sequences of words describing the topical content of a document. They are commonly called *"keywords"*, however the name causes confusion because they are *"multi-word terms"* or *"phrases"*. It is preferable to use *"keyphrases"* (Frank et al. 1999; Hammouda et al. 2005; Kim, Medelyan, et al. 2010; Peter D Turney 2000; Witten et al. 1999).

Kim, Medelyan, et al. (2010) defines *keyphrases* as the words that capture the main topics of a document and Frank et al. (1999) describes keyphrases as a *"high-level description of the content of a document's content that is intended to make easy to prospective readers to decide whether the document is relevant for them"*. Keyphrases can be assigned manually or extracted from the text body of the documents. Assigning *keyphrases* not present in

[6] http://cs.unibo.it/save-sd/2017/index.html
[7] https://scienceie.github.io

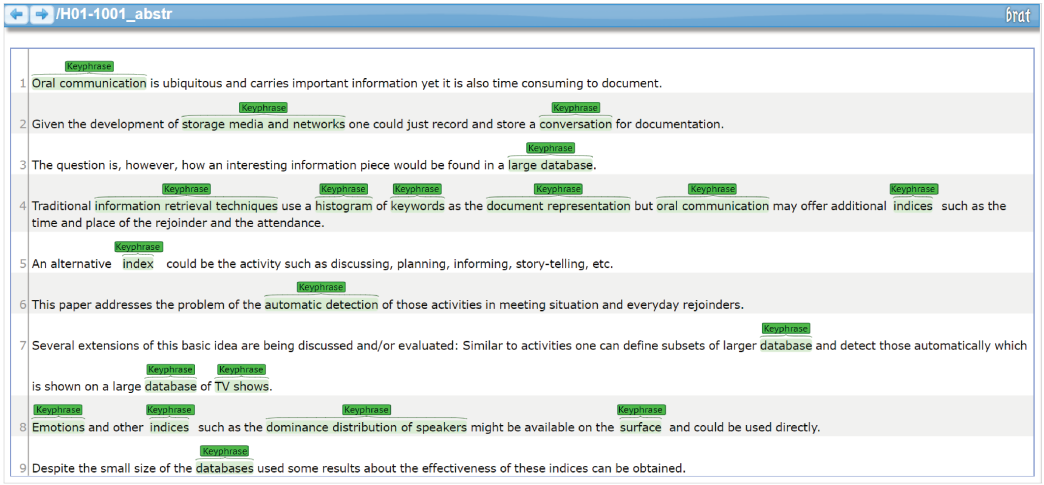
the text is out of the scope of this work, we only addressed the task of *automatic keyphrase extraction*.

The task of *automatic keyphrase extraction* is defined by Peter D Turney (2000) as *"the automatic selection of important, topical phrases from within the body of a document"*. The objective is to extract the key ideas related to the main topics in a document (Kim, Medelyan, et al. 2010; Kim, Medelyan, et al. 2013). Keyphrases are useful in tasks, such as text summarization (Mihalcea and Tarau 2004; Y. Zhang et al. 2004), document indexing (Gutwin et al. 1999), opinion mining (Berend 2011), they can be used for dimensionality reduction in text categorization (Hulth and Megyesi 2006; McCallum and Nigam 1999), document clustering (Hammouda et al. 2005) and assisting users in formulating queries (Gong and Q. Liu 2009; Gutwin et al. 1999).

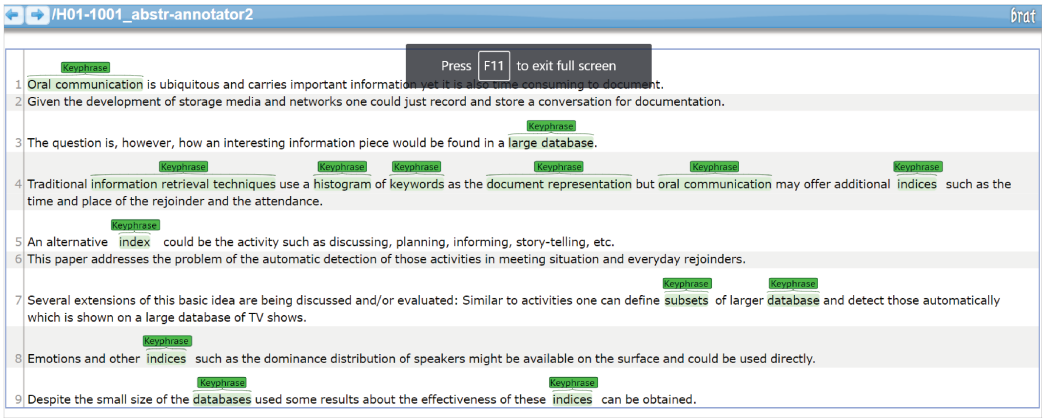
According to Frank et al. (1999), it is unfortunately that only a small number of documents have keyphrases assigned to them by the authors or the editors. Nowadays, we think it is a common practice by researchers to tag their articles with keyphrases in an effort to expand the diffusion of their works. However, there are issues associated to the assignation of keyphrases. For example, there are problems related to the semantics of the keyphrases (e.g. polysemy and synonymy). Or the criteria used to assign the keyphrases to a document, some keyphrases are oriented to improve the recall when looking for specific topics, but they misrepresent the content of the work. Usually only 10%-20% of the keywords occur in every document (Alexandrov et al. 2005; Pinto et al. 2006). Figure 1.2 shows an article's abstract annotated by two different persons. It illustrates the discordance of annotations, even on the most important keyphrases in the document.

1.2.1 Previous work

Keyphrase extraction is carried out with supervised, semi-supervised or unsupervised approaches. The goal of a supervised approach is to train a classifier with annotated texts. However, this type of resources are very complicated to generate, because annotating scientific papers with keyphrases requires a lot of specialization from human annotators. Some supervised techniques that are used to identify keyphrases are naïve Bayes, decision trees, boosting, maximum entropy, multi-layer perceptron, support vector machines (SVM), recurrent neural network (RNN) and conditional random fields (CRFs) (Augenstein et al. 2017; Hasan and Ng 2014; Kim, Medelyan, et al. 2010; Peter D Turney 2000). Unsupervised approaches generally are combinations of techniques based on clustering, graphs rankings or language models (Frank et al. 1999; Hasan and Ng 2014; Kim, Medelyan, et al. 2010; Mihalcea and Tarau 2004).



(a) *Annotator 1*



(b) *Annotator 2*

Figure 1.2 *Example of annotated keyphrases from two annotators in the same abstract. ACL-RD-TEC 2.0 abstract H01-1001 (QasemiZadeh and Schumann 2016).*

Filtering candidate keyphrases before training a model is a common approach, to reduce dimensionality of the data and propension to errors (Hasan and Ng 2014; Kim, Medelyan, et al. 2010; Peter D Turney 2000). There are different heuristics to accomplish this step, like, filtering noun phrases and prepositional phrases using regular expressions, fixed sets of rules or *part-of-speech tag sequences* (Haddoud et al. 2015; Hasan and Ng 2014).

1.3 Semantic similarity

A general definition of *semantic measures* is given by Harispe et al. (2015):

> *mathematical tools used to estimate the strength of the semantic relationship between units of language, concepts or instances, through a (numerical) description obtained according to the comparison of information supporting their meaning.*

However, this definition applies for *semantic relatedness* and *semantic similarity*. Then, specifically *measuring semantic similarity* on documents is the estimation of the degree in which the concepts of a document are similar to those in another one. It is expressed numerically with an scalar value, commonly in the range [0.0, 1.0], being 1.0 if they are the same and 0.0 meaning that they are completely different.

We could use different features to measure semantic similarity, like the lexical content that in a basic interpretation forms the meaning of the document according to *the distributional hypothesis* (Firth 1957; Harris 1954). To measure semantic similarity from the content, we might use words in n-grams, chunks, entities or paragraphs with different statistical methods to try to measure meaning.

Of course, metadata from the document is semantic information by itself, for example, keywords or categories given by the authors provide information about the domain of study. As well, we could assume that documents with the same authors are likely addressing related topics. A similar assumption applies for citations, because articles sharing citations are probably discussing the same concepts (Bolelli et al. 2006; Gollapalli and Caragea 2014; Nanba et al. 2011).

However, the metadata of the document is not enough, imagine for example the following hypothetical situation, a group of researchers in computer science writing an article about a chemistry problem in collaboration with a chemistry adviser. Imagine that they are addressing the chemistry problem using well known approaches in the field of computer science. Likely, the researchers are going to use some citations from their previous works and new

citations related to the problem that they are currently solving. Note that in the paper that they are writing the topics, the keywords, the citations and the indexing categories of the paper are going to be in both fields, chemistry and computer science. The readers of the resulting article (in the previous hypothetical example) might have different opinions about how it should be categorized, mostly depending on their personal perspective and criteria. It is an hypothetical case, but it is a very common situation adding complexity to the measure of similarity.

In the hypothetical case described above, the vocabulary used by the authors in the resulting document is likely more related to computer science than to chemistry, because of the background of the main writers, even if the main objective of the article is to describe a chemistry problem. This bias could affect how the document should be categorized or related to other documents addressing the similar concepts. In an opposite situation, when the article is redacted by a group of researchers in chemistry, we could assume that it is going to be written with a different perspective and vocabulary. Not only because of the professional background of the writers, simply, because it is another group of people influenced by different factors, i.e. geographical or temporal (Rangel et al. 2017).

We want to measure the semantic similarity on scientific publications. There are phenomena to consider while measuring the semantic similarity, like polysemy, words with multiple meanings, or synonymy, different words with the same meaning. In addition, there are adjacent problems to look at, for example, there are words commonly used across domains (*e.g., vector, variable, probability*) and any pair of documents containing those words are in some degree similar. The problem is that just because they share content it doesn't ensure that they are semantically similar. Thus, we should discriminate between lexically similar and semantically similar.

1.3.1 Previous work

The task of measuring semantic similarity on documents have improved with approaches such as LSA (Deerwester et al. 1990), Word2vec (Mikolov, Chen, et al. 2013; Mikolov, Sutskever, et al. 2013), Paragraph2Vec (Le and Mikolov 2014), GloVe (Pennington et al. 2014), FastText (Bojanowski et al. 2017), which are based on the idea that meaning depends on context (Firth 1957; Harris 1954).

The most effective methods at this moment are based on the generation of word embeddings with neural networks, i.e. Word2Vec, GloVe, FastText. Training models using those methods require a lot of computer resources, time and considerable amount of information, depending on the approach

taken, e.g. CBOW or Skip-gram to train Word2Vec. Note that the performance of those methods might vary depending on how the data is being used (Baroni, Dinu, et al. 2014). Variations to the word embeddings generated using neural networks consist in combinations of the previous methods with ontological resources like ConceptNet (Speer et al. 2017). There are also similar vector representations based on topic detection using LDA (Blei et al. 2003). Note that measuring semantic similarity with word embeddings is mostly applied on sentences and not in documents, e.g. (Brokos et al. 2016; Kusner et al. 2015).

1.4 Contributions

In our work on *automatic keyphrase extraction* we used *part-of-speech tags sequences* selected by their frequency in an human annotated corpus to filter candidate keyphrases and train a Conditional Random Field (CRF) model. Instead of filtering noun phrases or using sets of rules, as is commonly done, we used the PoS tag sequences from annotated Keyphrases and selected them in function of their frequency in the corpus.

We participated with the first version of our approach in "SemEval 2017 Task 10: ScienceIE - Extracting Keyphrases and Relations from Scientific Publications" (Augenstein et al. 2017). There were seventeen teams participating in the task and we ranked 11th, our results and the system description is presented in Hernandez et al. (2017a), the scenario of the task consisted in three subtasks, two of them were out of our scope, then we ranked 10th taking into account only the results for *keyphrase identification*. In the same system we included an approach for one of the other two subtasks, we used synsets from WordNet to classify keyphrases in three types (PROCESS, TASK and MATERIAL). We learned that there is a bias to the type PROCESS in the typed annotations of keyphrases in the dataset SemEval 2017 Task 10 (Buscaldi et al. 2017). It is described in Subsection 4.2.3 while the paper contains a wider description of the method used to extract the synsets and the analysis of the dataset. We also produced a list of PoS Tag sequences[8] from the training corpus, that we used to filter *candidate keyphrases* and the list of the synsets[9] that we used to classify the keyphrases. In Section 4.2 are described the preliminary results from this approach presented in Hernandez et al. (2017b). It includes the experimentation of using the filtered candidates

[8]PoS tag sequences from the training dataset https://github.com/sdhdez/corpus-data/blob/master/SemEval2017Task10/POSsequences.txt

[9]List of synsets from the training dataset https://github.com/sdhdez/corpus-data/blob/master/SemEval2017Task10/SynsetsRelatedToTrainingData.txt

and classifying keyphrases with SVM (Support Vector Machine) in which we found that there is an improvement in the keyphrase extraction using our approach. As part of the preliminary experimentation, we used different features to train different CRFs models and we produced a list of Regular Expressions[10] based on the PoS tag sequences to filter candidate keyphrases. The papers produced on this research are listed in Chapter 9.3.

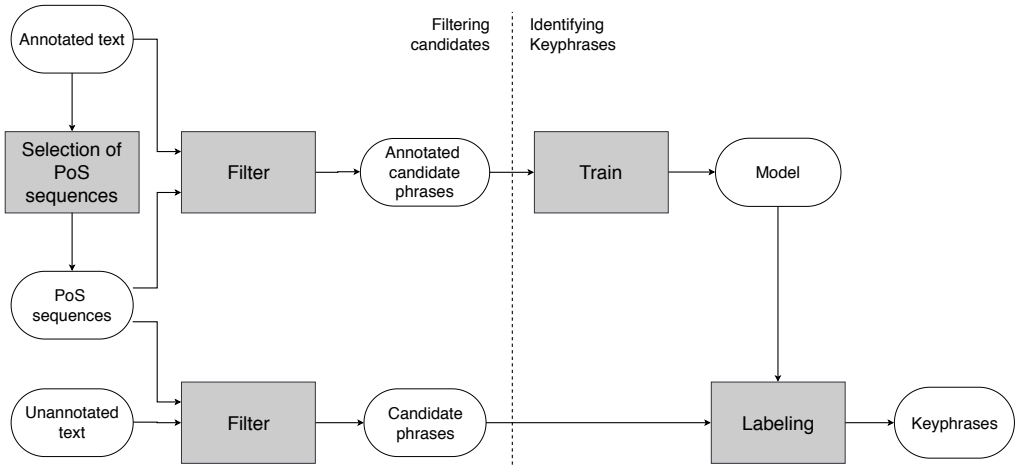


Figure 1.3 *Filtering candidate keyphrases to identify keyphrases using Part-of-Speech tag sequences.*

It is a more flexible method to filter keyphrases given that it can be adapted to any annotated corpus, in contrast with previous works based on predesigned general patterns. Since, the guidelines for the annotators change from corpus to corpus, this method might be useful to adapt the filtering to those variations.

We also proved that training CRFs models using candidate phrases instead of using the text as a whole improves the results of keyphrase extraction. Previous work showed that using the PoS tag sequences as a feature to train different models doesn't improve the results. However, we found that it is not true with all the tested corpora.

Based on our experiments we developed an Open-Source Python package. It is publicly available on GitHub [11] under the name *Kleis keyphrase extraction* [12], it is also available to install it using the Python package manager (*pip install kleis-keyphrase-extraction*, see Chapter 6 for a complete description).

[10]List of regular expressions from the PoS tag sequences https://github.com/sdhdez/corpus-data/blob/master/SemEval2017Task10/RegExpFromPOSsequences.dat

[11]GitHub https://github.com/

[12]Kleis package https://github.com/sdhdez/kleis-keyphrase-extraction

It is designed to facilitate the selection of PoS tag sequences and the filtering of candidate keyphrases from any annotated corpus. Currently, we included only four datasets, the SemEval 2017 Task 10, ACL-RD-TEC 2.0, SemEval 2010 Task 5 and Typed Entities from Tateisi et al. (2016). It includes CRFs models to label keyphrases, however the design allows to extend to other methods.

To our knowledge, there is not a corpus of scientific documents fit to evaluate semantic similarity between documents with annotated terms or keyphrases. To address this lack of resources, we selected a set of scientific abstracts from ArnetMiner and included annotations of keyphrases extracted with the python package Kleis (see Subsection 6 for a description). The extracted keyphrases match multi-word terms with multiple lexical representations along the set of abstracts. Those lexical representations in the abstracts are annotations of the same concept addressed with different names. It was achieved by matching Wikipedia redirections (article titles) [13] with the extracted keyphrases (see Section 8.5). This set of documents could be used to analyze the effectiveness of keyphrases as source of semantic information. We measured similarities on those abstracts, comparing different methods using words and keyphrases. We analyzed their correlations to observe how they perform in the presence of semantic information from keyphrases and the consequential loss of context.

1.5 Summary of chapters

This document is divided in three parts, Part I for the General Introduction, Part II for Automatic Keyphrase Extraction and Part III forSemantic Similarity.

[13] Wikipedia redirections https://en.wikipedia.org/wiki/Help:Redirect

Part II

Automatic Keyphrase Extraction

Chapter 2

Keyphrase extraction on scientific documents

Academic journals usually demand authors a list of keyphrases for their articles (Frank et al. 1999; Peter D Turney 2000) to facilitate the categorization and search of the document. *Keyphrases* are sequences of words representing the main topics of a document (Kim, Medelyan, et al. 2010; Peter D Turney 2000), they can be assigned manually or extracted from the document's content, in some cases they are chosen from a *"controlled environment"* (Witten et al. 1999). They are useful in tasks like text summarization, document indexing, text categorization, document clustering and queries suggestion (Frank et al. 1999; Kim, Medelyan, et al. 2010; Peter D Turney 2000).

The extraction of keyphrases from the body of a document is called *automatic keyphrase extraction* (AKE), there is recent interest in this task applied on scientific articles (Augenstein et al. 2017; Kim, Medelyan, et al. 2010), with aims to improve the search of the *state of the art*, recommend articles to readers, highlight missing citations to authors, identify potential reviewers for submissions, and analyze research trends over time.

It is not an easy task, the best result on keyphrase extraction in the SemEval 2017 Task 10 is $F_1 = 0.56$[1], as described by Augenstein et al. (2017),

> *"keyphrases are much more challenging to identify than e.g. person names, since* ***they vary significantly between domains****, lack clear signifiers and contexts and can consist of many tokens"*,

and generating resources to evaluate the task is not easy as is claimed by QasemiZadeh and Schumann (2016),

[1] https://scienceie.github.io/

> *"annotating terms and* ***building specialized vocabularies*** *is a much harder task than building resources for similar tasks such as named entity recognition".*

The main problem of the task is identifying the lexical boundaries of the keyphrases due to the relative criteria to define their *semantic relevance* across different domains of knowledge.

Recent work on keyphrase extraction showed improvements using supervised approaches, like naïve Bayes, decision trees, boosting, maximum entropy, multi-layer perceptron, support vector machines (SVM), recurrent neural network (RNN) and conditional random fields (CRFs) (Augenstein et al. 2017; Hasan and Ng 2014). However, given the difficulty of generating resources to experiment on specific domains, unsupervised approaches have been hard to outperform, applying combinations of techniques based on clustering, graphs, semantic networks, ranking and language models (Hasan and Ng 2014; Kim, Medelyan, et al. 2013).

In this chapter we detail the basis for the task of *automatic keyphrase extraction*, in Section 2.1 and 2.2 we define the concepts needed to understand the task, finally in Section 2.3 we describe the previous work on the matter.

2.1 Definition of *keyphrase*

In a general context, *keyphrases* are sequences of words describing the topical content of a document, in scientific literature they are commonly called *"keywords"*, however, this name is confusing, because very often keywords are *"multi-word terms"*. To avoid ambiguity with other equivalent denominations, i.e. *"key terms"*, *"key segments"* or *"key phrases"* (Frank et al. 1999; Grineva et al. 2009; Hammouda et al. 2005; Kim, Medelyan, et al. 2010; Peter D Turney 2000; Witten et al. 1999), we use the term *"keyphrases"* in the rest of the document.

There are different definitions of *keyphrases*, for example, Frank et al. (1999) describes keyphrases as

> *"a* ***high-level description*** *of the content of a* ***document's content*** *that is intended to make easy to prospective readers to decide whether the document is* ***relevant*** *for them"*

in the same manner, Witten et al. (1999) present keyphrases as individual entities of information,

> *"Keyphrases provide* ***semantic metadata*** *that* ***summarize*** *and* ***characterize*** *documents …they can be* ***interpreted individually*** *and independently of each other"*

while Peter D Turney (2000) defines keyphrases from another perspective, not individually, but in terms of a list of phrases on the topic of the document,

> *"a keyphrase list as a short list of phrases (typically five to fifteen noun phrases) that* ***capture the main topics*** *discussed in a given document …to quickly determine whether the given article is in the* ***reader's fields of interest****"*

Hammouda et al. (2005) make a distinction between keyphrases and keywords, but the key idea remains,

> *"A keyphrase is "a* ***sequence of one or more words*** *that is considered* ***highly relevant****", while a keyword is "****a single word*** *that is* ***highly relevant****." An arbitrary combination of keywords does not necessarily constitute a keyphrase; neither do the constituents of a keyphrase necessarily represent individual keywords."*

Kim, Medelyan, et al. (2010) describe keyphrases as

> *"****words*** *that capture the* ***main topics*** *of a document. As they represent these* ***key ideas*** *…keyphrases are useful as a form of* ***semantic metadata*** *indicating the* ***significance*** *of sentences and paragraphs"*

in a more recent work Haddoud et al. (2015) give a simpler definition

> *"A keyphrase is a* ***sequence of words*** *that* ***describes the content*** *of a document."*

In previous definitions there is a recurring principle, keyphrases are the *most relevant sequences of words* in the document. The perception of *"relevance"* of a keyphrase is relative to the *usefulness* for the reader and the semantic relation with the central topics in the document. This notion is defined as *phraseness* and *informativeness* by (Tomokiyo and Hurst 2003), *phraseness*

> *" is a somewhat* ***abstract notion*** *which describes the degree to which a given word sequence is considered to be a phrase"*,

and *informativeness*

> *"refers to how well a phrase* ***captures*** *or illustrates the* ***key ideas*** *in a set of documents"*,

each one depending on the criteria and perception of the readers for the target application.

2.1.1 Examples

Table 2.1 *Example of keyphrases (or keywords) in scientific articles provided by the author. Examples from the ACL-RD-TEC 2.0 and the Open Academic Graph.*

Article's title	*Provided keywords*
"Activity detection for information access to oral communication"	activity, dialogue processing, oral communication, speech, information access
"Isolation of Salmonella Enteritidis Phage Type 21b from a Eurasian Eagle-Owl (Bubo bubo)"	salmonella enteritidis phage type 21b, eurasian eagle owl, antimicrobial susceptibility, wild bird
"Image retrieval through qualitative representations over semantic features"	image retrieval, semantic gap
"Sorting and Selection with Imprecise Comparisons"	imprecise comparisons, noticeable difference unit, close-to-optimal solution, simple model, standard method, experimental psychology, large number, human subject, imprecise comparator, difference threshold, imprecise comparison

As shown in Table 2.1, keyphrases/keywords assigned by the authors of scientific articles can vary depending on the article, some are multi-word terms and others simple and common words (like "activity", "speech"), some are more descriptive than others and in some cases they are repetitive (like "imprecise comparisons", "imprecise comparator" and "imprecise comparison"). In those examples, it can be observed how the criterion to describe a document using keyphrases change depending on the author and the topic.

In Table 2.2 there is an example contrasting the keyphrases assigned by the authors of an article and the keyphrases annotated by Annotator 1 in the abstract of the same document, H01-1001 in ACL-RD-TEC 2.0 (QasemiZadeh and Schumann 2016). This last example makes a good illustration of how relevance, usefulness and informativeness are complex concepts depending on the perspective reader.

2.2 Definition of *keyphrase extraction*

Keyphrase extraction is a special case of *keyphrase generation* that can be categorized in two major approaches, *extraction* and *assignment* (Frank et

Table 2.2 *Example of the content of an article including keyphrase/keyword annotations provided by the authors/annotator. Includes occurrence counting for comparison. Article from the ACL-RD-TEC 2.0 abstract H01-1001 (QasemiZadeh and Schumann 2016)*

Article's title: "Activity detection for information access to oral communication"		
Type	*Keyphrases/keywords*	*Count*
Provided by author(s)	activity	20
	dialogue processing	0
	oral communication	12
	speech	17
	information access	10
Annotated (Annotator 1)	Oral communication	10
	storage media and networks	0
	conversation	9
	large database	4
	information retrieval techniques	0
	histogram	4
	keywords	5
	document representation	0
	indices	9
	index	4
	automatic detection	0
	database	18
	TV shows	4
	Emotions	9
	dominance distribution of speakers	0
	surface	0
	databases	4

al. 1999; Haddoud et al. 2015; Hasan and Ng 2014; Nguyen and Kan 2007; Peter D Turney 2000).

To clarify, Nguyen and Kan (2007) give a definition of both approaches, *keyphrase extraction*

> *"select phrases present in the source document itself ... usually consist of a* ***candidate identification stage*** *and a* ***selection stage****".*

and *keyphrase assignment*

> *"is typically used when the set of possible keyphrases is limited to a known, fixed set, usually derived from a* ***controlled vocabulary*** *or set of subject headings"*,

the latter, can be addressed as the task of *text categorization* (Frank et al. 1999), since it includes keyphrases not necessarily present in the text, this task is out of the scope of our current work.

In concrete, *automatic keyphrase extraction* is defined by Peter D Turney (2000) as

> *"the automatic selection of important,* ***topical phrases*** *from within the body of a document"*,

and by Hasan and Ng (2014) as

> *"is to extract a set of phrases that are related to the* ***main topics*** *discussed in a given document"*,

Despite there is a recurrent association of *topics* and *keyphrases* in the definitions, they are not considered to be equivalent, even though there is a relation. As stablished before, keyphrases are the key ideas related to the topics of a document (Hasan and Ng 2014; Kim, Medelyan, et al. 2010; Peter D Turney 2000; You et al. 2013), for that reason the task of *keyphrase extraction* has been used to address the task of *topic detection* (H. Li and Yamanishi 2000; Wartena and Brussee 2008).

There is intrinsic *relevance* and *informativeness* on the keyphrases in documents, making the task of *automatic keyphrase extraction* useful to improve many natural language processing (NLP) and information retrieval (IR) tasks (Hasan and Ng 2014), such as text summarization (Mihalcea and Tarau 2004; Y. Zhang et al. 2004), document indexing (Gutwin et al. 1999), opinion mining (Berend 2011), they can be used for dimensionality reduction in text categorization (S. Dumais et al. 1998; Hulth and Megyesi 2006; McCallum and Nigam 1999), document clustering (Hammouda et al. 2005) and in search engines dedicated to the academic domain, keyphrases can help to enhance the search of documents on large datasets or assisting users in formulating queries (Gong and Q. Liu 2009; Gutwin et al. 1999).

2.2.1 Task description

Keyphrase extraction is not an easy task due to the relative criteria to define the *semantic relevance* to identify the lexical boundaries of any possible keyphrase in the document. It usually addressed in two stages to extract keyphrases from the body of a document, *candidate filtering* and *keyphrase identification* (Hasan and Ng 2014; Nguyen and Kan 2007), as illustrated in Figure 2.1.

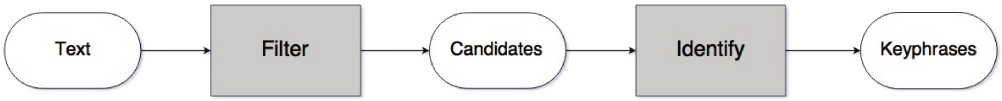


Figure 2.1 *Main stages in the task of automatic keyphrase extraction. Candidate filtering and keyphrase identification.*

Candidate filtering is selecting portions of the text from a document that are more likely to be keyphrases excluding the rest of the text. It is usually the first stage in systems for keyphrase extraction, however, it is optional and sometimes it is ignored completely. This commonly, this stage is addressed using heuristics (Haddoud et al. 2015; Hasan and Ng 2014).

Keyphrase identification in this stage, any portions of text are evaluated to decide if they are keyphrases or not. It is achieved with supervised and unsupervised approaches (Frank et al. 1999; Hasan and Ng 2014; Kim, Medelyan, et al. 2010; Peter D Turney 2000).

2.3 Previous work on keyphrase extraction

In this section we describe approaches related to *automatic keyphrase extraction*, focusing on works making *candidate filtering* using part-of-speech tags sequences and *keyphrase identification* using sequence labeling methods.

2.3.1 Candidate filtering using *PoS tag sequences*

Linguistic information helps to improve keyphrase extraction (Hulth 2003; Mihalcea and Tarau 2004), in specific, *part-of-speech tag sequences* or PoS tag sequences, achieve better results when used to filter candidate phrases (Frank et al. 1999; Haddoud et al. 2015; Hasan and Ng 2014; Kim and Kan 2009; Mihalcea and Tarau 2004). The problem lies on how to filter candidates more likely to be keyphrases, avoiding phrases that are *irrelevant* or phrases causing confusions. It is frequent to keep only *noun phrases* and *prepositional phrases* (Haddoud et al. 2015; Hulth 2003; Kim and Kan 2009; Mihalcea and Tarau 2004) given that the performance decreases when no exclusions are set (Haddoud et al. 2015; Mihalcea and Tarau 2004).

2.3.2 Keyphrase identification

The task of *identifying keyphrases* has been addressed with supervised and unsupervised approaches. The goal of a supervised approach is to train a classifier with annotated texts, however, this type of resources are very complicated to generate, because to annotate scientific papers with keyphrases requires a lot of specialization from human annotators. Some supervised techniques that are used to identify keyphrases are naïve Bayes (Frank et al. 1999; Witten et al. 1999), decision trees (Peter D Turney 2000), boosting, maximum entropy (Kim and Kan 2009), multi-layer perceptron, support vector machines (SVM), recurrent neural network (RNN) and conditional random fields (CRFs) (Augenstein et al. 2017; Hasan and Ng 2014). Unsupervised approaches generally are combinations of techniques based on clustering, graphs rankings or language models (Hasan and Ng 2014; Mihalcea and Tarau 2004).

Identifying keyphrases in a text document using *binary classification* approaches (Frank et al. 1999; Peter D Turney 2000; Witten et al. 1999) is denominated by Hasan and Ng (2014) as a *task reformulation*. The goal is to determine whether a candidate phrase is a *keyphrase* or a *non-keyphrases* (Hasan and Ng 2014). Keyphrases and non-keyphrases are used to train a classifier, although this approach is only helpful when the intention is to extract all the keyphrases in a text, otherwise a ranking approach is recommended.

Other very effective approach is considering the task a *sequence labeling* problem. Conditional Random Fields (CRFs) is used to address sequence labeling problems in natural language processing, such as, named-entity recognition (McCallum and W. Li 2003), identifying protein names in biology abstracts (Settles 2005), segmenting addresses in Web pages, finding semantic roles in text and citation extraction from research papers, symptom recognition (Holat et al. 2016) and others (Sutton and McCallum 2012). CRFs also have been used to address the task of automatic keyphrase extraction (Augenstein et al. 2017; Bhaskar et al. 2012; C. Zhang et al. 2008), this approach is typically implemented as shown in Figure 2.2.

Features for the previous approaches can be extracted from the documents, such as, statistical features (e.g., $tf - idf$, occurrences in a corpus), syntactic features (e.g., part-of-speech tag sequences, suffixes, prefixes) or structural features (e.g., position in the document). Also, external features could be included to improve the task performance, for example, ontologies (e.g., WordNet), the Web, knowledge bases (e.g., Wikipedia, log queries from search engines). An example of features used in our approach are shown in Table 2.3

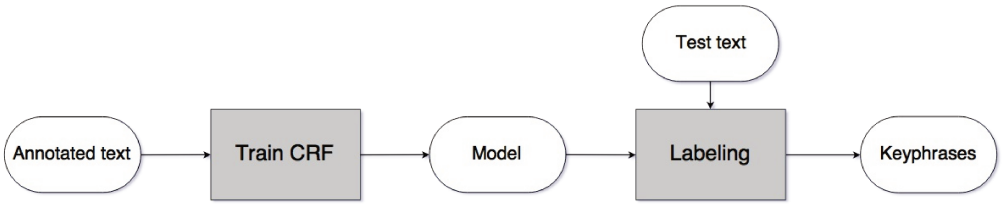


Figure 2.2 *Diagram representing the use of CRFs for the task of automatic keyphrase extraction.*

Table 2.3 *List of features used in our approach.*

Features
Word in lowercase
Part-of-speech
Two-character prefix of part-of-speech
One-character suffix
Two-character suffix
Uppercase (binary value)
Lowercase (binary value)
Title case (binary value)
Previous word in lowercase
Next word in lowercase
Beginning of the paragraph
End of the paragraph
One "bias" term

Recent work on automatic keyphrase extraction was presented at the *SemEval 2017 Task 10: Extracting Keyphrases and Relations from Scientific Publications* (Augenstein et al. 2017). This task was subdivided in three subtasks, Identification of keyphrases (Subtask A), Classification of identified keyphrases (Subtask B) and Extraction of relationships between two identified keyphrases (Subtask C). There were three scenarios to participate in this task, the only scenario including keyphrase identification is Scenario 1, which evaluation consist in solving the three subtasks. Results for keyphrase identification (Subtask A) are shown in table 2.4 [2].

As reported in the task description, seventeen teams participated in keyphrase extraction with different approaches and different levels of supervision. The best three teams used approaches based on recurrent neural networks (RNNs) and two of them have used a conditional random fields

[2]The full list of results can be found in https://docs.google.com/spreadsheets/d/1e6QPOxvxbo77cvAQfSEdwguVdA7vLngJAZy2TKjFMjQ/edit

Table 2.4 *Results by team on the task of keyphrase extraction at SemEval 2017 Task 10.*

Keyphrase extraction task	
Team	F_1 *score*
TIAL_UW	0.56
s2_end2end	0.55
PKU_ICL	0.51
TTI_COIN	0.5
NTNU-1	0.47
WING-NUS	0.46
SciX	0.42
IHS-RD-BELARUS	0.41
Know-Center	0.39
LIPN	0.38
SZTE-NLP	0.35
LABDA	0.33
NTNU	0.3
NITK_IT_PG	0.3
HCC-NLP	0.24
Surukam	0.24
GMBUAP	0.08

(CRF) layer on top of the RNNs, both works achieved a higher F_1 score for keyphrase identification compared to the other one.

Other approaches used by the rest of the teams are classification models based on random forest and support vector machines (SVM) with features such as $TF - IDF$ over a very large external corpus, IDF weighted word-embeddings, along with an existing taxonomy and noun phrase chunking. Several teams obtained a reasonable performance applying CRFs based methods with part-of-speech (POS) tagging and orthographic features such as presence of symbols and capitalization.

In particular, the ScienceIE task was focused on extracting keyphrases and relations between them, relying on the hypothesis that the ability of correctly recognizing these semantic items in text will help in tasks related to the process of scientific publishing, such as to recommend articles to readers, highlight missing citations to authors, identify potential reviewers for submissions, and analyze research trends over time. The hypothesis made by the organizers is that some concepts, notably PROCESS, TASK and MATERIAL, are cardinal in scientific works, since they allow to answer questions like: "which papers addressed a *Task* using variants of some *Process* ?". In their vision, *Processes* correspond to methods and equipment and *Materials* to corpora and physical items.

2.4 Datasets for keyphrase extraction

In this section we describe annotations in documents from available datasets to test the automatic keyphrase extraction.

2.4.1 Factors affecting extraction

Identifying keyphrases in scientific literature is different from other types of documents (Hasan and Ng 2014). Knowing this, we start by describing factors that affect the performance in task of automatic keyphrase extraction. According to Hasan and Ng (2014), there are four factors (or conditions) in text documents influencing how well keyphrases can be identified, i.e., the length of the document, structural consistency, topic change and topic correlation.

The impact of those factors variates depending on each type of text document, i.e., conversations, technical reports, narrations, etc. (Hasan and Ng 2014). We describe them in the context of scientific literature.

First, *length of the document*, the larger the document, the bigger the number of possible keyphrases. As described by Hasan and Ng (2010), the size of a document affects the size of a search space. Scientific documents contain hundreds of technicisms, each one of the might be a candidate phrase. However, in average scientific documents have around ten keyphrases.

Second, *structural consistency*, keyphrases tend to appear in certain locations in an structured document. It is an advantage for scientific documents, due to their structure, because keyphrases are more likely to be found in the introduction, the abstract or the conclusions.

Third, *topic change*, in conversational texts, topics change as the interaction moves forward. It is not the case for scientific documents, because they are written with to address a specific topic.

Fourth, *topic correlation*, it is an advantage for scientific documents, since keyphrases are commonly related between them, which is not the case for informal text. In an informal document, keyphrases appearing at the beginning of the document might not be related to the keyphrases at the end. The opposite occurs in scientific documents.

2.4.2 Datasets

We experimented with four datasets of scientific literature containing annotated keyphrases, i.e. SemEval 2010 Task 5 (Kim, Medelyan, et al. 2010), RANIS (Tateisi et al. 2016), ACL RD-TEC 2.0 (QasemiZadeh and Schumann 2016) and SemEval 2017 Task 10 (Augenstein et al. 2017).

All these datasets are publicly available. The annotations for the RANIS and SemEval 2017 Task 10 are distributed in the brat standoff format[3]. The raw text for each document is available in a text file (*.txt*) and the annotations in a brat file (*.ann*). The annotations for the ACL RD-TEC 2.0 is available in XML format. The the keyphrases for SemEval 2010 Task 5 are available in ranked lists without spans on the text.

SemEval 2010 Task 5 is a collection of papers from the ACM Digital Library with human annotated keyphrases. It contains 244 documents (144 for training and 100 for testing), from four evenly distributed research areas (distributed systems, information search and retrieval, multiagent systems and economics) (Kim, Medelyan, et al. 2010).

The dataset includes the full papers in raw text and a ranked list of keyphrases from the document.

```
C-45 : wireless sensor network,localization
C-46 : wireless sensor network,archival storage,index method
C-48 : dim,multi-dimensional range query
C-49 : opportunistic network,route,simulation
C-50 : search and rescue,sensor network
```

RANIS is a dataset for relational representation of context-dependent roles on information science papers. It is a corpus with 400 abstracts from the ACL anthology (230 for training and 20 for testing) and the ACM digital library (130 for training and 20 for testing) (Tateisi et al. 2016). It contains annotations of entities with ontology-based types and roles as relations on entities. Most of the annotations in this dataset are not keyphrases, however, it can be used for future comparison. A visualization of an annotated paragraph is shown in Figure 2.3. An example of the brat standoff format is shown in the next excerpt.

```
T1^^IPLAN-OR-PROCESS 0 35^^IMultilingual Coreference Resolution
T2^^IPLAN 69 100^^Imultilingual data-driven method
T3^^IQUALITY 64 67^^Inew
T4^^IPLAN-OR-PROCESS 105 127^^Icoreference resolution
T5^^IPLAN 150 157^^ISWIZZLE
. . .
R1^^IAttribute Arg1:T2 Arg2:T3^^I
```

[3] http://brat.nlplab.org/standoff.html

```
    R2^^ICoreference Arg1:T9 Arg2:T6^^I
    . . .
```

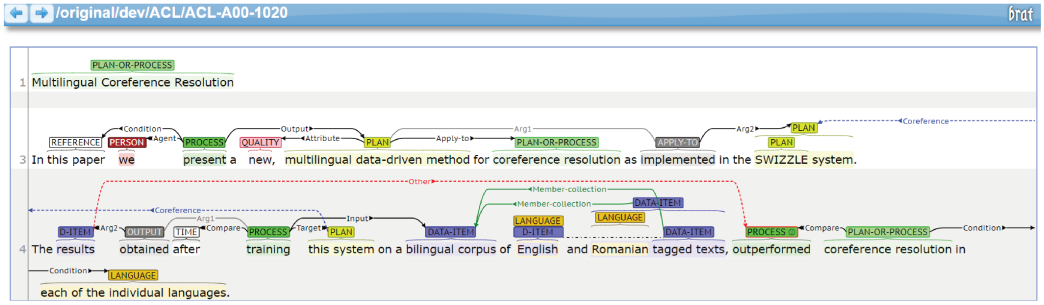


Figure 2.3 *Visualization of an annotated document from the RANIS dataset using the brat annotation tool.*

ACL RD-TEC 2.0 is a dataset consisting of 300 abstracts manually annotated from articles in the ACL Anthology Reference Corpus, published between 1978–2006 (QasemiZadeh and Schumann 2016). Single-word and multi-word terms are annotated and classified in seven categories. In this dataset there are 282 annotated files by the first annotator and 189 by the second. There is a total number of 171 abstracts with double annotations[4]. The manually annotated corpora for Annotator1[5] and Annotator2[6] can be browsed in the NoSkE engine, see Figure 2.4.

The annotations are provided in separated files depending on the annotator[7]. The raw text and the annotations are available in XML format as shown in the following example.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Paper acl-id="H01-1001">
    <Title>Activity detection for information access
    to oral communication</Title>
    <Section>
    <SectionTitle>ABSTRACT</SectionTitle>
        <S><term class="other">Oral communication</term> is ...
        <S>Given the development of <term class="tech">stor ...
```

[4]Official site for the ACL RD-TEC 2.0 dataset http://pars.ie/lr/acl_rd-tec
[5]http://pars.ie/lr/corpora/run.cgi/corp_info?corpname=aclrdtec2bq
[6]http://pars.ie/lr/corpora/run.cgi/corp_info?corpname=aclrdtec2ak
[7]Git repository for the annotations of the ACL RD-TEC 2.0 https://github.com/languagerecipes/acl-rd-tec-2.0

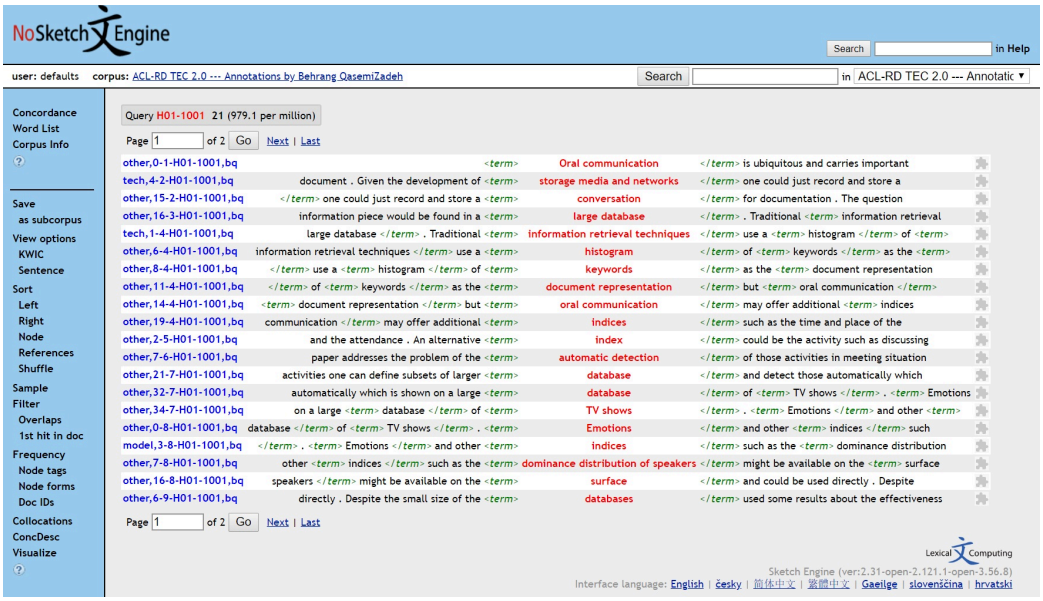


Figure 2.4 *Example of the NoSkE engine showing annotations from the ACL RD-TEC 2.0 dataset.*

```
        <S>The question is, however, how an interesting inf ...
        <S>Traditional <term class="tech">information retri ...
        <S>An alternative <term class="other"> ...
    </Section>
</Paper>
```

SemEval 2017 Task 10 corpus is built from ScienceDirect open access publications, 500 paragraphs from journal articles evenly distributed among the domains Computer Science, Material Sciences and Physics (Augenstein et al. 2017). It is formed of 350 documents for training, 50 for development and 100 for testing. It includes human annotations for keyphrases, each one labeled as MATERIAL, TASK or PROCESS. Additionally, there are annotated relations between keyphrases (synonyms or hypernyms), however they are out the scope of this work. An visualization of an annotated paragraph with the brat tool is shown in Figure 2.5. The example of the brat standoff format is shown in the next excerpt.

```
    T1^^IMaterial 0 5^^IWater
    T2^^IMaterial 180 185^^Iwater
    T3^^IMaterial 241 265^^Idistorted hydrogen bonds
    T4^^IProcess 383 419^^Iinner-shell spectroscopic techniques
```

```
T5^^IProcess 429 458^^IX-ray absorption spectroscopy
. . .
R1^^IHyponym-of Arg1:T1 Arg2:T24^^I
T25^^IProcess 460 463^^IXAS
*^^ISynonym-of T5 T25
R2^^IHyponym-of Arg1:T4 Arg2:T5
. . .
```

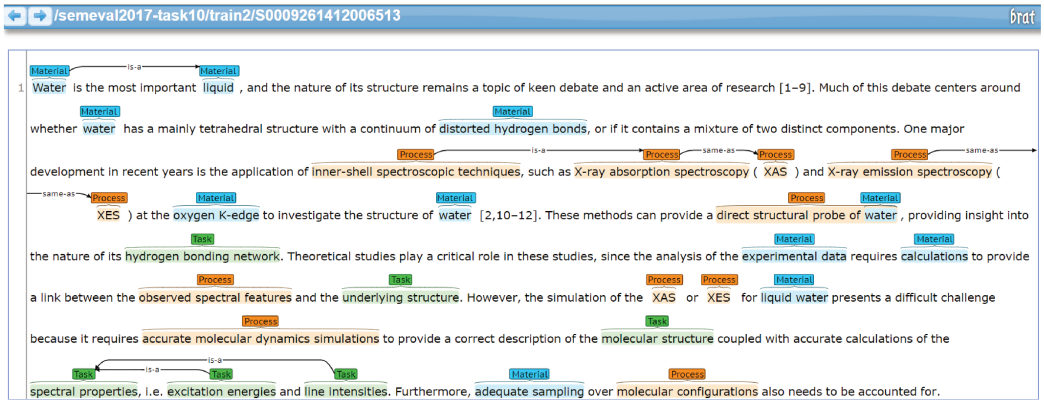


Figure 2.5 *Visualization of an annotated document from SemEval 2017 Task 10 dataset using the brat annotation tool.*

2.4.3 Evaluation of the datasets

To evaluate the performance of the automatic keyphrase extraction, the output of a system is matched exactly against the annotations. The output of a system usually includes an indication of the start and end of each keyphrase in a given text along the documents in the dataset. In other words, it is equivalent to the span in the brat format described previously. Given that all the spans from all the documents in a dataset are unique, each span is considered as an unique extracted keyphrase, even if the sequence of words in the text are repeated many times. The spans of the output are matched to the spans of the annotations. Only those exact span matches are considered to be the same keyphrase.

Given that K is the set of all the extracted keyphrases from all the documents of the dataset D. And A is the set of all the annotations in all the documents of the dataset D. An extracted keyphrase is a triplet of the form $(d, start, end) \in K$ and an annotated keyphrase is $(d, start, end) \in A$. Given that $(d, start, end)$ is an element of K or A. Where $start, end$ are the starting

position and ending position of a keyphrase in the document d, given that $d \in D$.

The number of extracted keyphrases matching exactly the annotations are defined as *true positives*, TP. It is the cardinality of the intersection of both sets, A and K.

$$TP = |K \cap A|$$

The number of extracted keyphrases that do not match annotations are considered *false positives*, FP. It is the cardinality of the relative complement of A in K.

$$FP = |K - A|$$

The annotations that do not match extracted keyphrases are *false negatives* FP. It is the cardinality of the relative complement of K in A.

$$FN = |A - K|$$

They are used to calculate the micro-averaged *precision* (p_{ma}), *recall* (r_{ma}) and *F-score* (F_β where $\beta = 1.0$). Which are defined as follows:

$$p_{ma} = \frac{TP}{TP + FP}$$

$$r_{ma} = \frac{TP}{TP + FN}$$

$$F_{1.0} = 2 \cdot \frac{p_{ma} \cdot r_{ma}}{p_{ma} + r_{ma}}$$

Observe that these micro-averaged measurements are calculated along all the documents in the dataset and they are equivalent to their common and more used versions of precision and recall. We specify this measurements to avoid confusion with the macro-averaged version of the measurements.

Chapter 3

Our approach on keyphrase extraction

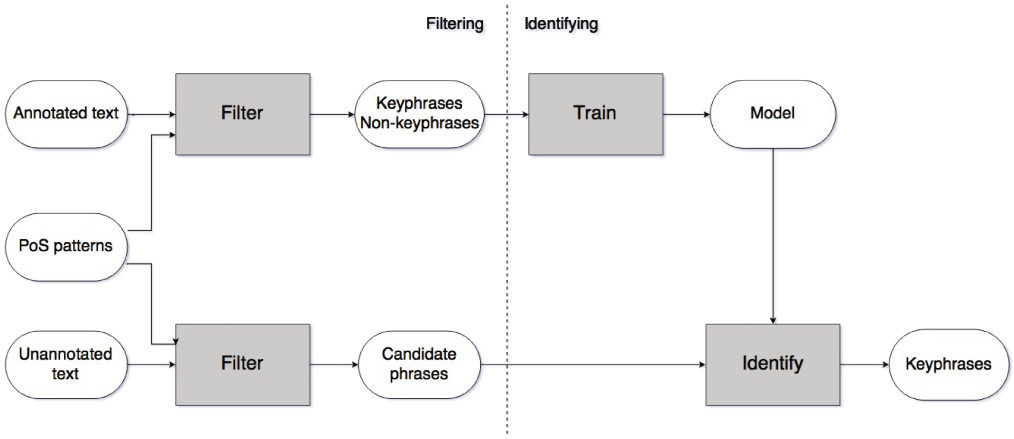


Figure 3.1 *Overview of our approach.*

In this chapter we present our approach on the automatic extraction of keyphrases from scientific documents. In our work we used *part-of-speech tag sequences*, or *PoS tag sequences*, selected by their frequency in an human annotated corpus to filter candidate keyphrases, those candidates are used to train a model to identify keyphrases. The approach is based in the assumption that keyphrases in scientific documents are mostly constructed following certain part-of-speech patterns, giving that the language in these kind of documents is formal.

We address the keyphrase identification stage as a binary classification problem (Frank et al. 1999), hence the two possible classes are *keyphrases* and

non-keyphrases. In Figure 3.1 is shown a general overview of our approach, it is based on the stages described in Subsection 2.2.1.

3.1 Part-of-speech tag sequences

The *part-of-speech tag* of a single word is its grammatical category represented by a label, it describes the function of the word in the structure of a sentence, e.g., *NN* indicates a noun, *VB* a verb and *RB* an adverb. A *part-of-speech tag sequence*, or *PoS tag sequence*, is an arrangement of tags representing a grammatical structure.

"The/DT accuracy/NN of/IN the/DT prediction/NN and/CC
detection/NN models/NNS was/VBD 96.6/CD %/NN and/CC 84.1/CD
%/NN ,/, respectively/RB ./."

Figure 3.2 *Example of sentence tagged with part of speech. Excerpt of text from the testing dataset in the SemEval 2017 Task 10 (Augenstein et al. 2017).*

To tokenize and tag all the examples in this work we used the TreebankWordTokenizer[1] and the PerceptronTagger[2], both from NLTK 3.4[3], respectively. The tags of part of speech are those in the pre-trained model for the PerceptronTagger, the CoNLL data consisting of the WSJ part of the Penn Treebank.

In Figure 3.2 there is an example of a tagged sentence[4], for each token in the sentence there is a tag of part of speech representing its function in the sentence. Observe that *%* symbols are tagged as nouns, but they could be rarely considered a keyphrase. In Figure 3.3 is shown the full annotated text of the example.

The whole *part-of-speech tag sequence* for the sentence in Figure 3.2 is

"DT NN IN DT NN CC NN NNS VBD CD NN CC CD NN , RB .".

The arrangement of tags from a segment of the same sentence is also a PoS tag sequence, e.g., *NN NNS* is the sequence for *detection models*, which is an example of a *noun phrase*.

[1] TreebankWordTokenizer in the NLTK package https://www.nltk.org/api/nltk.tokenize.html#nltk.tokenize.treebank.TreebankWordTokenizer
[2] PerceptronTagger in the NLTK package http://www.nltk.org/api/nltk.tag.html#nltk.tag.perceptron.PerceptronTagger
[3] Official site for the NLTK package https://www.nltk.org/
[4] File S0885230816301759.txt from the testing dataset in SemEval 2017 Task 10.

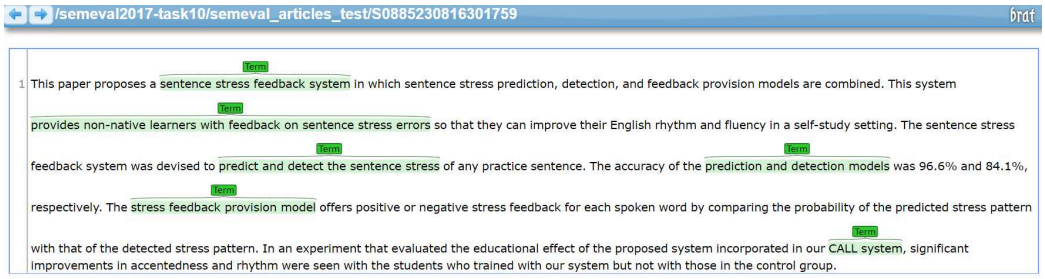


Figure 3.3 *Annotated abstract from the testing dataset at SemEval2017 Task10 (Augenstein et al. 2017).*

3.2 PoS tag sequences for candidate filtering

In the stage of candidate filtering (described in Subsection 2.3.1) for the task of automatic keyphrase extraction the common idea is to filter segments of text more likely to be keyphrases, those segments are called *candidate keyphrases*.

PoS tag sequences are commonly used as patterns to filter candidates, matching predefined tag arrangements in segments of the target text, e.g., matching *noun phrases* or *prepositional phrases*. The segments of text matching the sequences are the filtered candidate keyphrases. See Table 3.2 for an example.

The performance of extracting keyphrases using *PoS tag sequences* in the filtering stage vary depending on the filter used to match candidates, the impact of the use of this kind of linguistic information is compared by Haddoud et al. (2015), in their work they implemented a system using *logistic regression* as learning algorithm. They tested five filters of PoS tag sequences reported in related works. Those filters are shown in Table 3.1, they appear in the form of regular expressions. Haddoud et al. (2015) remarks that, even though the filters are predefined to filter *noun phrases*, the definition of a noun phrase vary from one work to another.

The first filter in the Table 3.1 is the proposed by Haddoud et al. (2015), the regular expression satisfies the definition of noun phrases according to their observations of the training data, it also achieves the best performance with their system for keyphrase extraction. Unlike the filters for other systems, their proposal includes cases of noun phrases containing verbs at the past participle, i.e., the tag *VBN*, such as *"multi-agent distributed system"* (*JJ VBN NN*) or *"unified framework"* (*VBN NN*).

The list in the bottom of Table 3.2 shows an human annotated keyphrase and the largest noun phrases recovered using the regular expressions from Table 3.1. The sentence is an excerpt from the paragraph shown in Figure 3.3.

Table 3.1 *Regular expressions matching noun phrases (Haddoud et al. 2015).*

Related works	*Filter*
Haddoud et al. (2015)	(NN|NNS|NNP|NNPS|JJ|VBN|NN IN|NNS IN)* (NN|NNS|NNP|NNPS|VBG)
P. Turney (1997), Pal et al. (2011)	(NN|NNS|NNP|NNPS|JJ)*(NN|NNS|NNP|NNPS|VBG)
Kim and Kan (2009), Kim, Medelyan, et al. (2010)	NBAR IN NBAR where NBAR = (NN|NNS|NNP|NNPS|JJ|JJR|JJS)*(NN|NNS|NNP|NNPS)
F. Liu et al. (2009)	(JJ)*(NN|NNS|NNP)+
Nguyen and Kan (2007)	(NN|NNS|NNP|NNPS|JJ|JJR|JJS)*(NN|NNS|NNP|NNPS)

Table 3.2 *Example of an human annotated keyphrase and filtered candidates with Haddoud et al. (2015). Sentence annotated from the testing dataset of SemEval 2017 Task 10.*

"The/DT accuracy/NN of/IN the/DT prediction/NN and/CC detection/NN models/NNS was/VBD 96.6/CD %/NN and/CC 84.1/CD %/NN ,/, respectively/RB ./.

Text segment	*PoS tag sequence*	*Type*
"prediction and detection models"	NN CC NN NNS	Human annotated
"accuracy"	NN	Filtered
"prediction"	NN	Filtered
"detection models"	NN NN	Filtered
"%"	NN	Filtered
"%"	NN	Filtered

It is to be observed that the human annotated keyphrase is not retrieved with those filters, because it includes a conjunction (*CC*) between two noun phrases. It is an example of the lost of candidates as consequence of filtering only noun phrases, however, the most *relevant* words are kept and a different person could assert that the correct keyphrases should be those that were filtered. This kind of variations are frequent in the SemEval corpus and other corpus, because of the disagreement of the annotators and the pre-defined directives to annotate.

3.2.1 Selection of the PoS tag sequences

PoS tag sequences are often selected to filter candidates based on linguistic observations or statistical experiments of the corpus, trying to maximize the retrieve of keyphrases, while excluding patterns more likely to cause errors.

To pick the PoS tag sequences, first, we extracted the sequences from the annotated keyphrases in the training dataset of the SemEval 2017 Task 10 (Augenstein et al. 2017). We obtained in total 1445, the 60 most frequent [5] are shown in Table 3.3, sorted in decreasing order by their occurrence in the corpus. The list in Table 3.4 shows a portion of the PoS tag sequences occurring only one time in the corpus.

Table 3.3 *First 60 PoS tag sequences ordered by number of occurrences. Extracted from the training dataset in SemEval 2017 Task 10*

Count	*PoS tag sequence*	*Count*	*PoS tag sequence*	*Count*	*PoS tag sequence*
1333	NN	23	NNP NN NN	13	CD NN
559	NN NN	22	JJ NN NN NN	12	VBG NN NNS
414	JJ NN	21	NNP	12	VBG DT JJ NN
301	NN NNS	21	NN IN NN	12	NN IN NNS
293	NNS	20	VBG NN NN	12	JJ NNP NN
289	JJ NNS	19	NNP NNP NNP	12	JJ JJ NN NN
200	JJ NN NN	19	JJ NNP	11	NNP JJ NN
151	NNP NN	18	NN IN DT NN	11	NN IN NN NNS
138	JJ NN NNS	17	NNS NN	10	VB DT JJ NN
94	NN NN NN	16	RB NN	10	NN VBG NN
90	NNP NNS	16	NN VBG	10	NNP NNP NN
61	VBG	16	JJ NNP NNS	10	NN IN JJ JJ NNS
54	NN NN NNS	16	JJ NN NN NNS	9	VB DT NN NN
52	JJ JJ NN	15	NNP NNP	9	RB JJ NNS
51	JJ	15	NNP NN NNS	9	NN NNP
41	VBG NN	14	NN JJ NNS	9	NN NN NN NN
32	JJ JJ NNS	14	**NN CC NN NNS**	9	NN IN NN NN
31	VBN NN	13	VBN JJ NN	9	NN IN JJ NN
28	VBN	13	NN JJ NN	9	NN IN DT JJ NN
28	VBG NNS	13	JJ CC JJ NNS	9	JJ NN IN NN
⋮	⋮	⋮	⋮	⋮	⋮
25	NN IN JJ NNS	13	DT NN		
⋮	⋮	⋮	⋮		
23	VBN NNS	13	CD NNS		

We observed from the extracted PoS tag sequences, (see Table 3.3,) that not all the noun phrases are similarly frequent e.g., *NN*, *NN NN*, *NN NN NN NN*, in fact, not all the keyphrases can be retrieved by the definition of noun phrase in which the filters in Table 3.1 are based, e.g. *NN IN DT NN*, *NN CC NN NNS* or *JJ CC JJ NNS*. Other cases are more frequent than simpler noun phrases, e.g., *JJ NN NN NN*, *NN IN NN*. Like *VBN*, i.e. the verb in past participle, or *VBG NNS*, i.e. the verb in present participle followed by a plural noun. In Table 3.4 there are sequences occurring one time in the corpus, thus less likely to occur, presumably, as consequence of the parentheses or the length of the segment of text.

The diversity of PoS tag sequences in human annotated keyphrases could

[5] Extracted PoS tag sequences available in https://github.com/sdhdez/corpus-data/blob/master/SemEval2017Task10/POSsequences.txt

Table 3.4 *Example of PoS tag sequences occurring one time in the training dataset in SemEval 2017 Task 10.*

PoS tag sequence	*PoS tag sequence*
CD CC CD NN NNP	CD NNP IN CD
CD CC CD NNS	CD NNS IN JJ NNS
CD IN NN NNS	CD NNS IN NNP NNS
CD JJ JJ NN	CD VBG
CD JJ NN	DT
$ CD JJ NN *NNP* NN NN	DT JJ JJ
CD JJ NN NNS	DT JJ JJ NN IN DT NN NN NN
CD JJ NN VBN *NNP* NNP NN NN	DT JJ NN IN DT NN
CD *NN*, *NN* CD NN	DT JJ NN IN DT NN NNS IN DT NN
CD NN NN IN DT JJ NN	DT JJ NN IN NN NN NNS
CD NN NNP	DT JJ NN JJ JJ IN JJ NN
CD NN NNP , CD NN NNP CC CD NN NNP	DT JJ NN NN CD NN
CD NN *NNP* IN *NNPNNPCD*	DT JJ NNP
CD NN NNP JJ NNS	DT JJ NNP IN CD NNS
CD NN *NNP* NN	DT JJ NNP VBN NN

be attributed to a bad annotation or to errors in PoS tagging, even though, the definition of keyphrase is very open and does not make restrictions of this kind, as long as the keyphrase keeps relevance and usefulness for the reader, see Section 2.1. Actually, the PoS tag sequence of the human annotated keyphrase in the example of Table 3.2 (testing dataset), is shown in Table 3.3 (training dataset).

Motivated by the previous observations, our assumption is that most frequent sequences are more likely to be keyphrases. In our opinion, it exposes the alternative of picking the PoS tag sequences by their frequency in a given corpus. Therefore, we used each PoS tag sequence as an unique pattern to filter candidates, instead of using restrictive patterns, as proposed in other approaches.

We also generated a set of 1225 regular expressions [6] from the PoS tag sequences in Table 3.3, for example, the sequences *JJ NN*, *JJ NN NN*, *JJ JJ NN* and *JJ NN NN NN NN* were collapsed into a single regular expression *(JJ)+ (NN)+*. Part of the set of regular expressions is shown in Table 3.5.

The regular expressions match the largest candidate keyphrases, on the contrary to the PoS tag sequences. For example, if there is a sequence of two nouns in the text *NN NN*, the PoS tag sequences match the part of speech of

[6] Generated PoS tag sequences available in https://github.com/sdhdez/corpus-data/blob/master/SemEval2017Task10/RegExpFromPOSsequences.dat

Table 3.5 *Regular expressions generated from the PoS tag sequences in the training data. Ordered by number of matches in the training dataset from SemEval 2017 Task 10.*

Matches	*RegExp*	*Matches*	*RegExp*
1995	(NN)+	37	VBN (NN)+
704	(JJ)+ (NN)+	35	(NN)+ IN (NN)+
364	NN (NNS)+	29	JJ (NNP)+
327	(JJ)+ NNS	29	(VBG)+ NNS
299	(NNS)+	28	VBN
195	NNP (NN)+	27	(NN)+ IN DT (NN)+
162	(JJ)+ NN NNS	24	VBN (NNS)+
94	(NNP)+ NNS	23	(NNS)+ NN
64	(VBG)+ NN	23	(NN)+ JJ (NN)+
62	(VBG)+	20	(NNP)+ (NN)+ NNS
58	(NNP)+	19	JJ (NNP)+ (NN)+
57	(JJ)+		⋮

three segments of text (i.e., *NN NN*, *NN*, *NN*), and the regular expressions match only a single segment of text (i.e. *NN NN*). The first set of patterns produces more candidates and overlapping than the second.

3.2.2 Filtering candidates

We used the two sets of patterns, in Table 3.3 and 3.5, to filter candidate keyphrases. It is done by matching each PoS tag sequence, see Subsection 3.2.1, with the part-of-speech sequences in the target text, all possible matches are considered filtered candidates.

> "provides/VBZ an/DT approach/NN to/TO circumvent/VB the/DT sign/NN problem/NN in/IN numerical/JJ simulations/NNS"

Figure 3.4 *Segment of tagged text from the development dataset from SemEval 2017 Task 10.*

Take as example the excerpt of text in Figure 3.4 from the development dataset in Augenstein et al. (2017), the human annotated keyphrases are highlighted and it includes the part of speech of each word. The PoS tag sequences for "*sign problem*" and "*numerical simulations*" are *NN NN* and *JJ NNS*, respectively. Those can be filtered with the filters addressed by

Haddoud et al. (2015) and using the PoS tag sequences in Table 3.3. The highlighted segments in the example are not the only matches in the excerpt, Table 3.6 list all the filtered candidate keyphrases (and the highlighted keyphrases), obtained by matching the PoS tag sequences selected from the training corpus.

Filtering candidate phrases with n-grams, i.e. from unigrams to 5-grams, in segments of text from Figure 3.4, produces 45 different candidates, which are more than the number of candidates shown in Table 3.6. It is an example of the dimensionality reduction of candidates.

Table 3.6 *Filtered candidates using the PoS tag sequences. Ordered by counts of the PoS tag sequences.*

Counts	*PoS tag sequence*	*Candidate keyphrase*
1333	NN	approach
1333	NN	sign
1333	NN	problem
559	NN NN	sign problem
293	NNS	simulations
289	JJ NNS	numerical simulations
51	JJ	numerical
25	NN IN JJ NNS	problem in numerical simulations
13	DT NN	an approach
13	DT NN	the sign
9	VB DT NN NN	circumvent the sign problem
6	VB DT NN	circumvent the sign
5	DT NN NN	the sign problem
2	VBZ DT NN	provides an approach
2	NN NN IN JJ NNS	sign problem in numerical simulations
2	NN IN	problem in
2	NN IN JJ	problem in numerical
1	DT	an
1	NN TO VB	approach to circumvent
1	DT	the

If the filtering were made using only the PoS tag sequences occurring at least 14 times, observe the middle line in Table 3.6, then the reduction of candidate keyphrases is notably greater, without losing the annotated keyphrases, at the same time that it excludes irrelevant words to understand the topic of the text, e.g., *an, to, circumvent, the, provides*, although, it is expected a reduction of the retrieved keyphrases.

The previous observation motivates our criteria used to pick the PoS tag sequences for candidate filtering. In the experimentation chapter is shown how the number occurrences of each PoS tag sequence in the training corpus

affect the quality of the candidates based on the performance to identify keyphrases.

3.3 Keyphrase identification

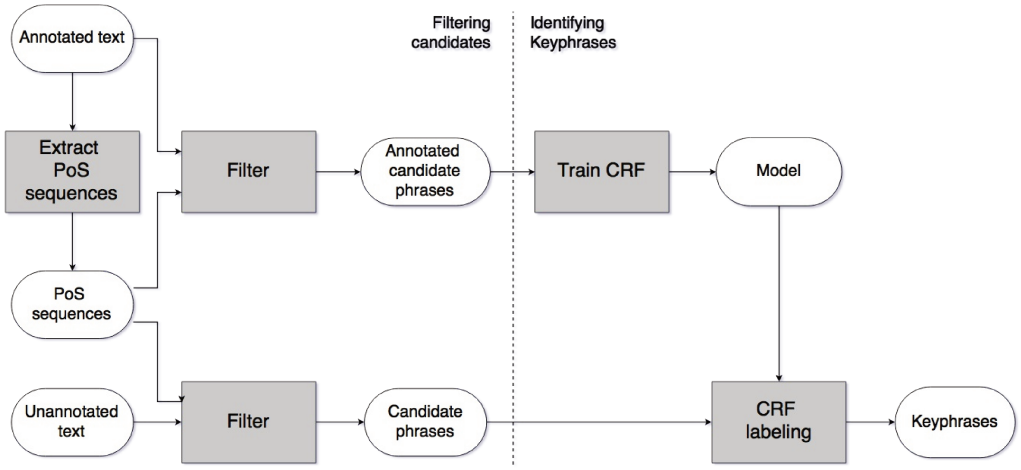


Figure 3.5 *Filtering candidates with PoS tag sequences to train a CRF model to label keyphrases.*

To address the stage of *keyphrase extraction* we trained a Conditional Random Field model (CRF model) to label keyphrases, as shown in Figure 3.5. The CRF model was trained with the candidate keyphrases described in Subsection 3.2.2, each candidate keyphrase is treated as an independent sentence. We used orthographic features, such as part of speech, suffixes and the capitalization of the words, see features in Table 2.3.

3.3.1 Features

To train our CRF model we used the features suggested in the documentation of *python-crfsuite*[7] for the task of named entity recognition[8], we didn't make a deep analysis of them. For each word in the target text we extracted the features listed in Table 2.3. An example of the features in a text segment is shown in Table 3.7.

[7] https://python-crfsuite.readthedocs.io/en/latest/

[8] Example of *python-crfsuite* https://github.com/scrapinghub/python-crfsuite/blob/master/examples/CoNLL%202002.ipynb

Table 3.7 *Example of features extracted from a segment of the text in Figure 3.4.*

Features	in	numerical	simulations
Word in lowercase	in	numerical	simulations
Part-of-speech	IN	JJ	NNS
Two-character prefix of part-of-speech	IN	JJ	NN
One-character suffix	n	l	s
Two-character suffix	in	al	ns
Uppercase (binary value)	False	False	False
Lowercase (binary value)	True	True	True
Title case (binary value)	False	False	False
Previous word in lowercase	problem	in	numerical
Next word in lowercase	numerical	simulations	of
Beginning of the paragraph			
End of the paragraph			
One "bias" term	bias	bias	bias

3.3.2 Training models

We extracted all the possible candidate keyphrases from the training corpus, using all the PoS tag sequences described before, see Subsection 3.2.1.

Table 3.8 *Filtered candidate keyphrases using the PoS tag sequences. Ordered by counts of the PoS tag sequences.*

PoS tag sequence	*Candidate keyphrase*	*Label*
NN	approach	NON-KEYPHRASE
NN	sign	NON-KEYPHRASE
NN	problem	NON-KEYPHRASE
NN NN	sign problem	KEYPHRASE
NNS	simulations	NON-KEYPHRASE
JJ NNS	numerical simulations	KEYPHRASE
JJ	numerical	NON-KEYPHRASE
NN IN JJ NNS	problem in numerical simulations	NON-KEYPHRASE
DT NN	an approach	NON-KEYPHRASE
DT NN	the sign	NON-KEYPHRASE
VB DT NN NN	circumvent the sign problem	NON-KEYPHRASE
VB DT NN	circumvent the sign	NON-KEYPHRASE
DT NN NN	the sign problem	NON-KEYPHRASE
VBZ DT NN	provides an approach	NON-KEYPHRASE
NN NN IN JJ NNS	sign problem in numerical simulations	NON-KEYPHRASE
NN IN	problem in	NON-KEYPHRASE
NN IN JJ	problem in numerical	NON-KEYPHRASE
DT	an	NON-KEYPHRASE
NN TO VB	approach to circumvent	NON-KEYPHRASE
DT	the	NON-KEYPHRASE

Here we introduce the term "*annotated candidate phrases*" to characterize filtered candidate keyphrases matching exactly an human annotated keyphrase. These *annotated candidate phrases* are labeled as KEYPHRASE

in Table 3.8, the rest of the candidates are labeled as NON-KEYPHRASE. Both labels are used like in a binary classification problem (Frank et al. 1999).

Figures 3.6 and 3.7 show candidate keyphrases filtered using PoS tag sequences. Both candidates include context words from the text they were extracted. The highlighted text represents an annotation in the dataset. The labels for each word represents the BIO notation. In this notation each word is labeled using **B**, **I** or **O**, depending if the word is in the **b**eginning, **i**nside or **o**utside of the candidate. In our approach, the context words are labeled as **O** because they are not part of the candidate.

"the/O sign/B problem/I in/O"

Figure 3.6 *Example of a KEYPHRASE with two context words in BIO notation.*

Note that in Figure 3.6 there is highlighted text and it is not the case in Figure 3.7. The difference between both candidates is that the second is not annotated in the training corpus. It is shown to emphasize that not all candidate keyphrases are annotated in the dataset.

"the/O sign/B problem/I in/I numerical/I simulations/I of/O"

Figure 3.7 *Example of a NON-KEYPHRASE with two context words in BIO notation.*

Other notation tested in our work is the BILOU notation, i.e., **b**eginning, **i**nside, **l**ast, **o**utside or **u**nit. Figure 3.8 shows an example with the previous candidate.

"the/O sign/B problem/I in/I numerical/I simulations/L of/O"

Figure 3.8 *Example of a NON-KEYPHRASE with two context words in BILOU notation.*

Training a CRF model with candidate phrases

Note that a CRF model is typically trained with a full sentence or segments of text delimited by a given window from an annotated text, for example, the annotated text in Figure 3.9 is a single input to train the model [9].

[9] This example was modified to show only keyphrases, the original annotations include the labels TASK, PROCESS and MATERIAL. For this reason there are overlapped annotations of keyphrases.

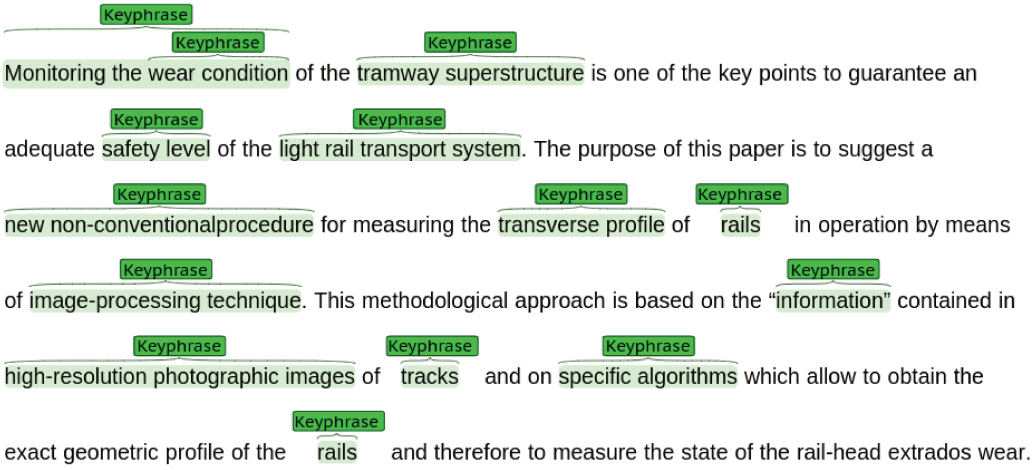


Figure 3.9 *Annotated document from the training dataset in SemEval 2017 Task 10. Showing only keyphrases.*

Instead, we make the assumption that each candidate keyphrase is an independent input. Our CRF model is trained using the candidate keyphrases filtered from the training dataset (Augenstein et al. 2017) with the BIO notation. Each candidate keyphrase is processed as an independent text and labeled as KEYPHRASE or NON-KEYPHRASE, like the segments shown in Table 3.6. The filtered candidates in Figure 3.10 are examples of the inputs used to train our CRF model.

3.3.3 Labeling candidates

We identify keyphrases in an unannotated corpus by labeling candidate keyphrases from the target text, using a CRF model trained as described in Subsection 3.3.2. The candidate keyphrases are filtered with the PoS tag sequences, as explained in Subsection 2.3.1. Each candidate keyphrase is considered a single input to label with the CRF model, with the mentioned features in Subsection 3.3.1.

As we mentioned, in contrast with a typical implementation of CRFs, we consider each candidate keyphrase as a single input. The problem of labeling using this approach is that it produces overlapped keyphrases. We keep the largest keyphrase if case there is overlapping of identified keyphrases.

3.3.4 Preliminary experimentation

We made preliminary experimentation using filtered candidates with variations of the approach presented in this document. Preliminary results were

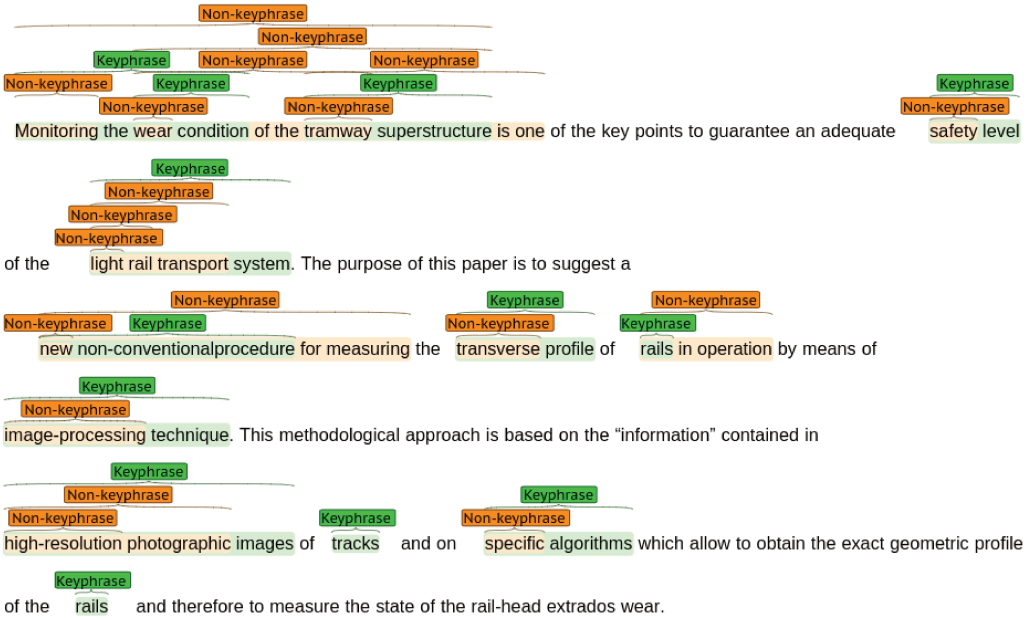


Figure 3.10 *Candidate keyphrases (KEYPHRASES and NON-KEYPHRASES) in a document from the training dataset in SemEval 2017 Task 10.*

used to address the subtasks at SemEval 2017 Task 10. We experimented with additional features, support vector machines (SVM) and a different labeling notation in the training phase. Additionally, there are changes in the pre-processing and post-processing steps.

Labeling notation

In the preliminary experimentation, we used BIO notation to label input to train CRF models. However, in subsequent experiments we found that BILOU notation improves performance.

Feature variations

We experimented using additional features to address the subtasks at SemEval 2017. Those features were extracted from a database of academic papers and WordNet. Note that the preliminary experiments do not include the PoS tag sequence as a feature. Later in this document, we show that training a CRF model using PoS tag sequences improves the performance of the keyphrase extraction.

Titles from academic papers. We used information from titles in academic papers. Titles have been previously useful in the task of automatic keyphrase extraction (Grineva et al. 2009; Hasan and Ng 2014).

We generated a database of bigrams, trigrams and the part of speech of the trigrams, extracted from titles in academic papers from the Microsoft Academic Graph[10]. Our approach is focused to the English language. We excluded the non-English papers using the Python tool *guess_language*[11]. The English titles were PoS tagged and inserted in a database as bigrams and trigrams.

We added four binary features for each token in a candidate keyphrase. The value of the binary features (True or False) depends on whether the n-grams formed with the token context exist or not in the database. For example, we included four binary features for the token "*sign*" in Figure 3.4, since it forms the n-grams "*the sign*", "*sign problem*", "*the sign problem*" and *DT NN NN*. We found that these features do not produce a significative improvement on performance.

Features for keyphrase type classification. We added WordNet-based features to the previously discussed features. To classify the keyphrases as a subtask of the SemEval 2017 (Augenstein et al. 2017). The dataset includes annotations of the type of the keyphrase. The idea is to classify keyphrases in three different types PROCESS, TASK and MATERIAL. Our participation in the subtask of keyphrase classification is reported in Hernandez et al. (2017a). Additional experiments and analysis are reported in Buscaldi et al. (2017).

WordNet (George A Miller 1995) is a well known lexical database for the English language. In WordNet, word senses are represented as *synsets*, or "set of synonyms", which may be connected to other synsets by some relationship. Some of the most common relationships are meronymy (part-of) and hypernymy (is-a). We define a *synpath* as the list of synsets connecting a sense of a target word to the root of the hierarchy in WordNet, following the hypernymy relation.

In Figure 3.11 are shown the synpaths corresponding to the three senses of the word *extraction* in WordNet 3.0. The definitions of the senses are as follows:

1. extraction#1:the process of obtaining something from a mixture or compound by chemical or physical or mechanical means;

[10]Version 2016/02/05 https://academicgraphwe.blob.core.windows.net/graph-2016-02-05/index.html
[11]https://pypi.python.org/pypi/guess_language-spirit

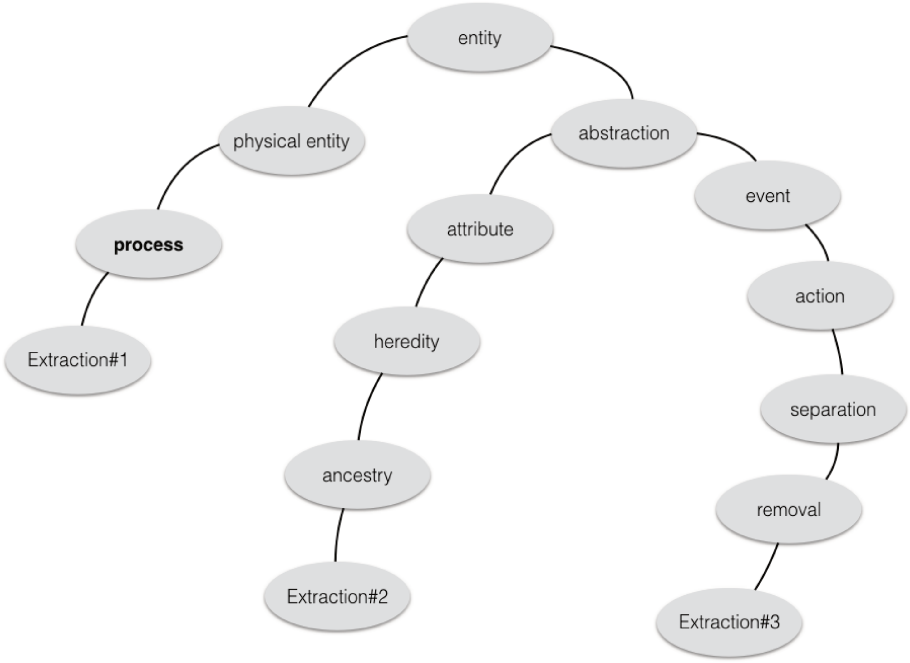


Figure 3.11 *Example of synpaths for the word "extraction" in WordNet 3.0 (simplified by removing some synsets).*

2. extraction#2: properties attributable to your ancestry;

3. extraction#3: the action of taking out something (especially using effort or force).

From Figure 3.11 it can be observed that the synset *process* is in the synpath (process, physical_entity) of `extraction#1`, which seems an important clue to classify this keyword as a PROCESS, according to the ScienceIE classification. Therefore, we supposed that synpaths can be effectively used as features to predict the category of a keyword. Given the number of synsets in WordNet (more than $117,000$), we opted to select only a subset of those synset, in particular by limiting the scope to the synsets that are particularly distinctive for each of the three classes.

We calculated, on the training corpus of SemEval 2017, the probability $p(s|C)$ for each synset with respect to class C. Subsequently, we ordered in decreasing order, for each class, the synsets according to the difference $p(s|C_i) - \frac{p(s|C_j)+p(s|C_k)}{2}$.

We show in Table 3.9 the most distinctive synsets for each category. The semantic correlation between the MATERIAL category and its distinctive synsets is particularly evident.

Table 3.9 *Top 5 distinctive synsets for each category.*

PROCESS	*MATERIAL*	*TASK*
psychological_feature.n.01	physical_entity.n.01	science.n.01
event.n.01	object.n.01	possession.n.02
abstraction.n.06	whole.n.02	natural_science.n.01
act.n.02	artifact.n.01	question.n.02
cognition.n.01	matter.n.03	subject.n.01

We arbitrarily selected the top 20 distinctive synsets for each category as binary features[12]. These features are true for a token in a keyphrase if they are present in any of the hypernym paths connecting the noun synsets to the root synset. Notice that these features were added only for the nouns, since there is no hierarchy for other lexical categories (excluding verbs, whose hierarchy is very shallow in comparison to nouns). If the keyphrase is compound by multiple words, only are used the synpaths for the rightmost noun.

SVM

In our preliminary experimentation, we used Support Vector Machine (SVM) models trained with filtered candidate keyphrases. Each candidate keyphrase is converted to a word vector of TF-IDF[13], i.e., Term Frequency - Inverse Document Frequency, using the words in the candidate. It means that the $tf - idf$ of each word is a feature. There is a model for each threshold of filtered candidates. A class is represented by the centroid of all the vectors made of candidate keyphrases of the same label. There are four classes, one for each type of keyphrase (e.g., *non-keyphrases*, *process*, *material* and *task*) (Augenstein et al. 2017).

Pre-processing

The pre-processing step in the preliminary experimentation vary from the one used in the final approach. We used different methods to tag annotated keyphrases with their part-of-speech. This variation changes the resulting

[12]The synsets are available in https://github.com/sdhdez/corpus-data/blob/master/SemEval2017Task10/SynsetsRelatedToTrainingData.txt

[13]TF-IDF is better described in Subsection7.2.1

PoS tag sequences, therefore the filtering changes. In the preliminary experimentation, we PoS tagged keyphrases directly from the annotations without including context tokens. In the final approach, we PoS tagged directly the full abstracts. This simple change affects the resulting set of PoS tag sequences and the results.

Post-processing

Subphrases. Given that every candidate keyphrase is an input to be labeled by the CRF model, there is a chance that inside large candidates labeled as non-keyphrases could be a subsegment of text labeled as a valid keyphrase. We extracted those keyphrases to increase the performance in recall. However, we found in subsequent experiments that it produces a negative effect in the final $F - score$. We excluded this step in the final approach.

Exclusion list. We generated an exclusion list of words with the intension of improving the precision by removing identified keyphrases that are frequent along the training dataset, e.g. *system*, *data*. This list contains single-word terms from the training corpus, whose inverse document frequency (idf) is lesser or equal than a threshold. The threshold is the mean of idfs from all the tokens in the dataset minus four times their standard deviation.

Chapter 4

Experimental results

In this chapter there is a description of experiments on the task of automatic keyphrase extraction. First, an analysis of our baseline is shown. It is followed for a description of the preliminary results that motivated our approach (described in Chapter 3). After, results of variations of our approach are presented, i.e., variations in features, post-processing, selection of the PoS tag sequences.

4.1 Baseline

In this section, we analyze our baseline experiments on automatic keyphrase extraction. They are used as point of comparison to the results of our approach. We use the development dataset from SemEval 2017 Task 10 (Augenstein et al. 2017) to compare the results.

All the results are shown as values of *precision*, *recall* and the $F_1.0$ measure.

4.1.1 Filtered candidates

To start, we defined a reference point to compare the results of our experiments. We measured the *precision*, *recall* and $F_1.0$ score of all the candidate keyphrases filtered from the development dataset. We used the PoS tag sequences from the training corpus in SemEval 2017, see Subsection 2.3.1.

The results shown in Table 4.1, correspond to the evaluation of all the filtered candidate keyphrases, in total $14,905$. The $recall = 0.84$, is the highest target result. The lost of 16% correspond to the annotated keyphrases with a PoS tag sequence not present in the training dataset. However, we think it is an acceptable lost given the dimensionality reduction compared

Table 4.1 *Evaluation of filtered candidates as keyphrases. Development dataset from SemEval 2017.*

precision	*recall*	F_1
0.065	0.84	0.12

to all the possible candidates.

The results in Figure 4.1 show the evaluation of the candidate keyphrases filtered with the PoS tag sequences selected by their numbers of occurrences. Each data point in Figure 4.1 represents a subset of the PoS tag sequences. Each subset is selected by excluding sequences with a threshold in the number of occurrences of the annotated keyphrases from the training dataset. For the *recall*, the first data point represents the evaluation of candidates filtered using all the PoS tag sequences. The second point correspond to candidates filtered with a subset of PoS tag sequences, excluding those occurring one time in the training dataset. The third point is obtained with the subset of sequences excluding those with less than 3 occurrences. It continues in the same manner for the rest of the sequences. The last point is obtained from single-words nouns, because *NN* is the only PoS tag sequence with 1333 occurrences.

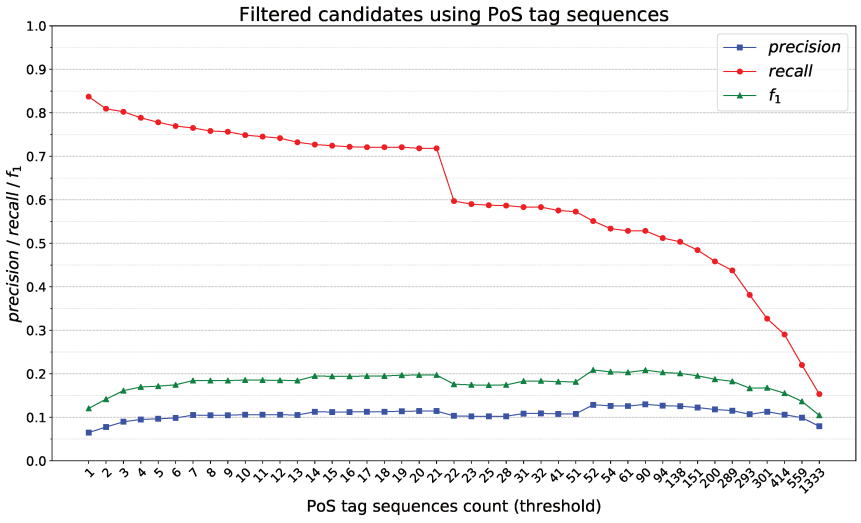


Figure 4.1 *Filtered candidates from the development dataset using PoS tag sequences from the training dataset.*

In the same graph, Figure 4.1, there is an evident drop in recall in the

change from using PoS tag sequences with 21 and 22 occurrences. This change is because of the lost of the PoS tag sequences "NNP" and "NN IN NN", since both appear 21 times in the training data, see Table 3.3.

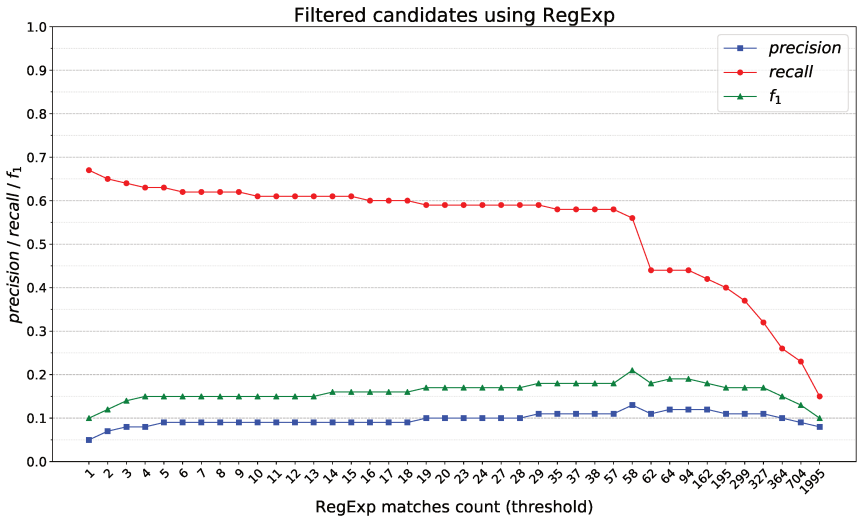


Figure 4.2 *Filtered candidates using regular expressions based on PoS sequences.*

The results shown in Figure 4.2 correspond to the evaluation of the filtered candidate using the regular expressions, see example in Table 3.5. In the figure is applied the same principle to select the subsets of regular expressions, using the number of matches of keyphrases in the training dataset.

Observe that it shows a similar drop in *recall* with the change from 58 to 62 matches in the training data, it is because the lost of the regular expression matching any keyphrase with the PoS tag sequence *"NNP"*, e.g., singular proper nouns.

4.1.2 CRF model (baseline)

Our baseline is the CRF model trained using only the annotated keyphrases from the training dataset in the SemEval 2017 corpus. The training is made as described in Subsection 3.3.2, using the BIO notation, with the features listed in Table 3.7 and without candidate filtering. The results are shown in Table 4.2. We use this baseline to compare performance for the rest of the experiments. The Figure 4.3 shows a comparison between the baseline and the filtered candidates.

Table 4.2 *Baseline. Results of the CRF model trained without candidate filtering.*

	precision	*recall*	F_1
All filtered candidates	0.065	0.84	0.12
CRF model	0.46	0.31	0.37

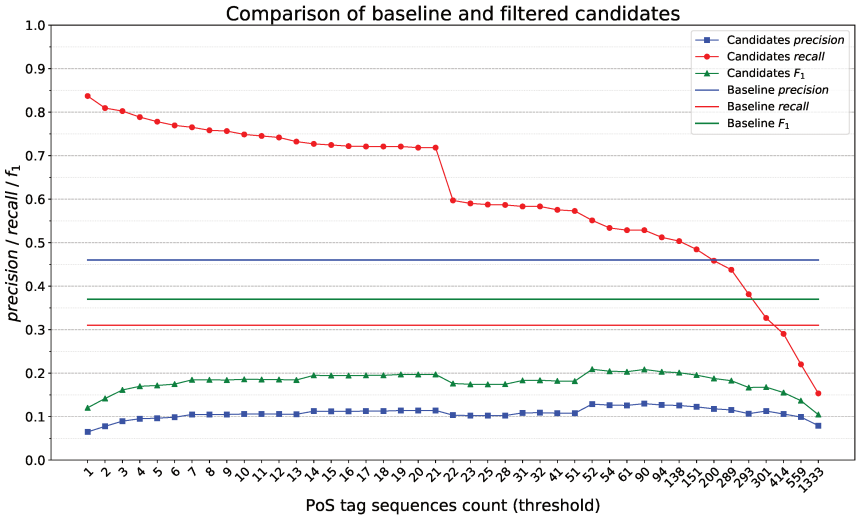


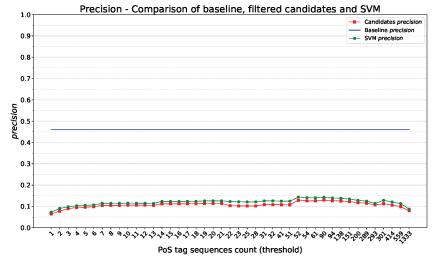
Figure 4.3 *Comparison of the baseline and the filtered candidates.*

4.2 Preliminary experimentation

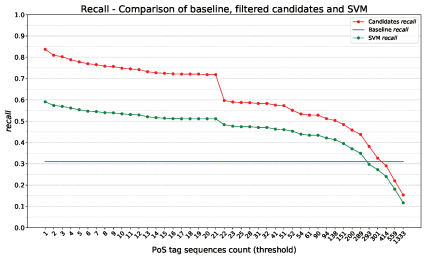
In this section we describe the preliminary experiments to the approach presented in this document.

4.2.1 SVM using typed keyphrases

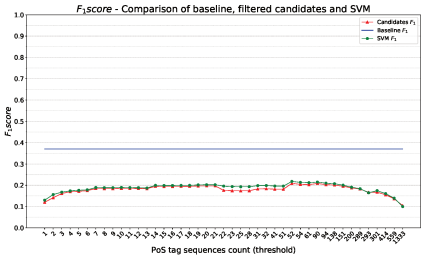
In this experiments, see Figure 4.4, we show the evaluation of keyphrase extraction using a Support Vector Machine (SVM) model trained with candidate keyphrases. The F_1 score of this method does not outperform the baseline. However, it can be observed that the recall performance is directly affected by the filtered candidates used to train each model. This observation motivated our approach.



(a) *Precision.*



(b) *Recall.*



(c) $F_1 score$.

Figure 4.4 *Evaluation of keyphrase extraction with SVM trained using candidates and type of keyphrase. Choosing largest keyphrase.*

4.2.2 Experiments with CRF

In this section we describe different experiments of CRF models trained with filtered candidates using PoS tag sequences. These experiments are previous variations of our approach.

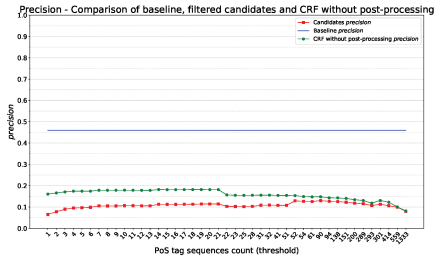
Post-processing

Here we describe the results of the experiments with variations in the post-processing step.

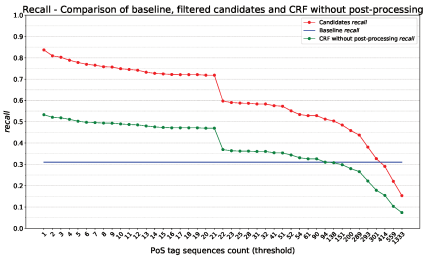
CRF model without post-processing. In this experiment there is not post-processing step, then there is overlapping in the resulting keyphrases giving a similar behavior to the evaluation of the filtered candidates, see Figure 4.5. We labeled the filtered candidates using a CRF model. It was trained using annotated keyphrases including the type of the keyphrase from the training data, the resulting classes are *non-keyphrase, process, material and task.*

It has the best performance in *recall.* However, it is barely better in *precision* than the direct use of the candidates. It does not reach better

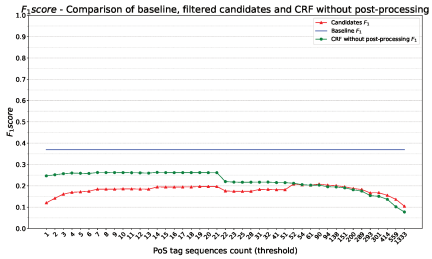
results than the baseline and does not represent the best performance of a CRF model trained with filtered candidates.



(a) *Precision.*



(b) *Recall.*

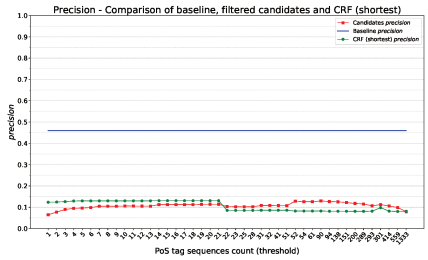


(c) F_1 *score.*

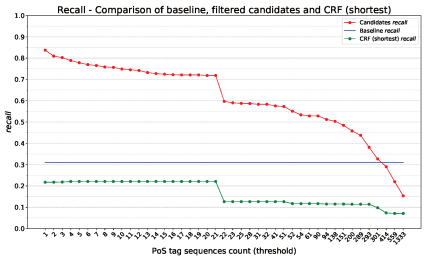
Figure 4.5 *Evaluation of keyphrase extraction using a CRF model trained with filtered candidates.*

Shortest keyphrase. In this experiment we tested the removal of overlapping in the output of the CRF model trained with the filtered candidates, from Subsection 4.2.2. It is done by keeping the shortest keyphrases when there are two overlapped keyphrases in the output. For example, assuming that the following segments of text are extracted keyphrases, *"sign problem in numerical simulation"* and *"numerical simulations"*, only the later is kept as output because it is the shortest. It presents the worst performance overall. See Figure 4.6.

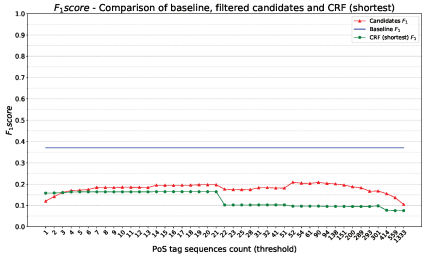
Largest keyphrase In this experiment we tested the removal of overlapping in the output of the CRF model trained with the filtered candidates, see Subsection 4.2.2. In this case we kept the largest keyphrase when there is overlapping. For example, assuming that the following segments of texts are keyphrases, *"sign problem in numerical simulations"* and *"numerical simulations"*. Only the first is kept as output, because it is the largest. This step improves the performance and reaches one of the best F_1 score of our experiments. See Figure 4.7.



(a) *Precision.*

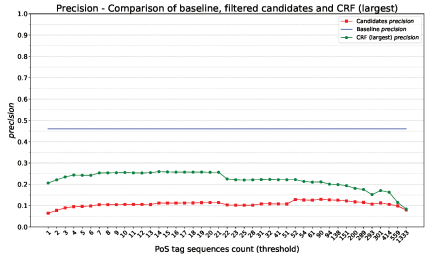


(b) *Recall.*

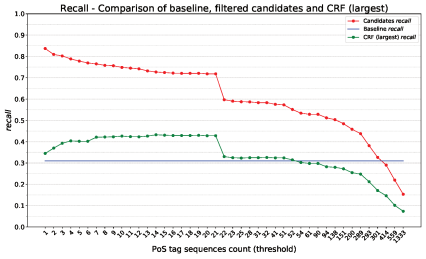


(c) $F_1 score$.

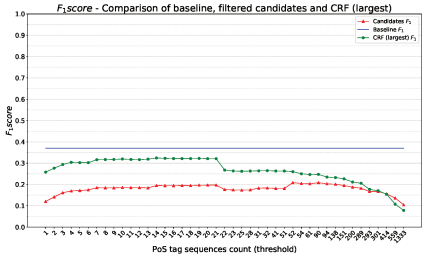
Figure 4.6 *Evaluation of keyphrase extraction using a CRF model trained with filtered candidates. Keeping the shortest keyphrase.*



(a) *Precision.*



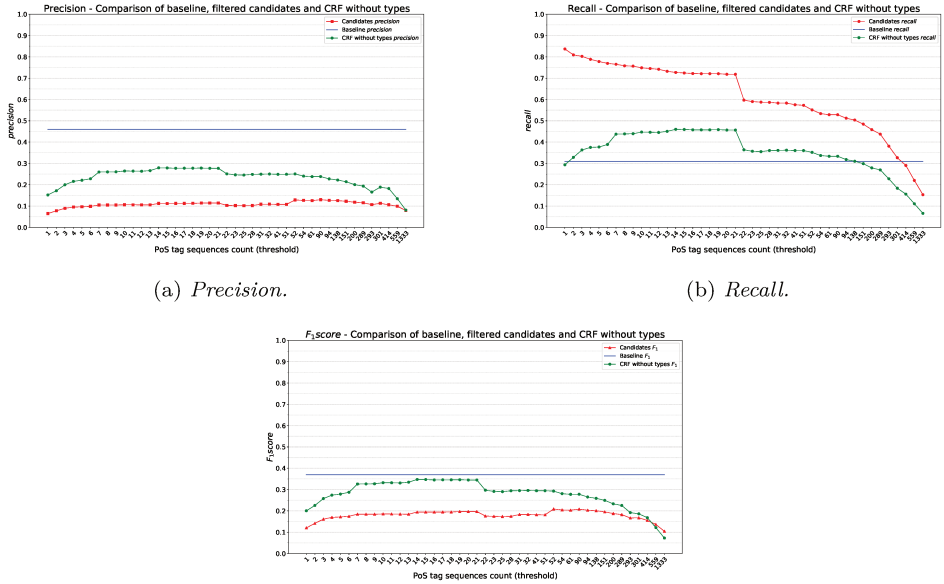
(b) *Recall.*



(c) $F_1 score$.

Figure 4.7 *Evaluation of keyphrase extraction using a CRF model trained with filtered candidates. Keeping the largest keyphrase.*

CRF model using filtered candidates without type In this experiment we used a trained CRF model as described before. We used two labels, NON-KEYPHRASE and KEYPHRASE. The removal of the overlapping is made by choosing the largest keyphrases. It reaches the best performance in F_1 score from the preliminary experiments. Also, it is the base of the approach presented at SemEval 2017 Task 10 and in this work. Observe Figure 4.8.



(a) *Precision.* (b) *Recall.*

(c) F_1 *score.*

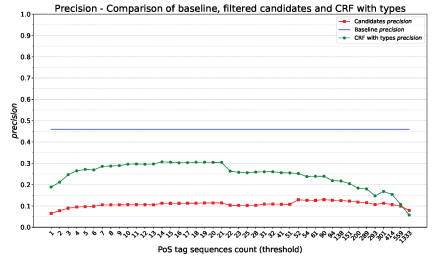
Figure 4.8 *Evaluation of keyphrase extraction using a CRF model trained with filtered candidates and without types of keyphrases. Keeping the largest keyphrase.*

4.2.3 Type classification

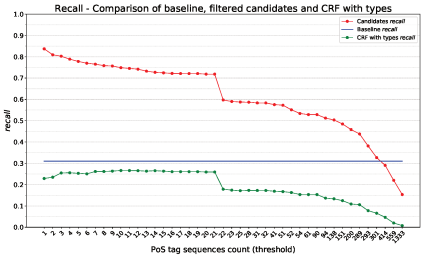
The SemEval 2017 Task 10 corpus (Augenstein et al. 2017) includes the type of keyphrase, representing one of the following classes, PROCESS, TASK or MATERIAL. We experimented using those labels instead of the described KEYPHRASE and NON-KEYPHRASE. However, addressing the task of *type classification* is not the main goal of this work.

Training model with the type of keyphrase. In this case we used CRFs as described in the approach. Instead of training one model to label all the types, we trained three models to label whether a candidate keyphrase is of a specific type, e.g., *process*, *material* and *task*. The removal of overlapped

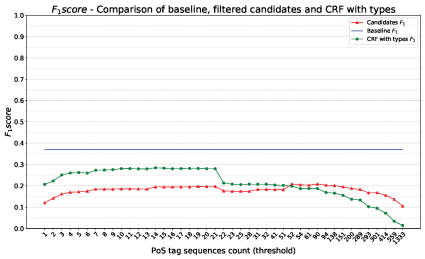
keyphrases is made by choosing the largest keyphrases, like described before. This experiment reaches the best *precision* of the preliminary experiments using filtered candidates. However there is a meaningful drop in recall. It doesn't outperforms the results of a typical implementation of the baseline in F_1 score. See Figure 4.9.



(a) *Precision.*



(b) *Recall.*



(c) F_1*score.*

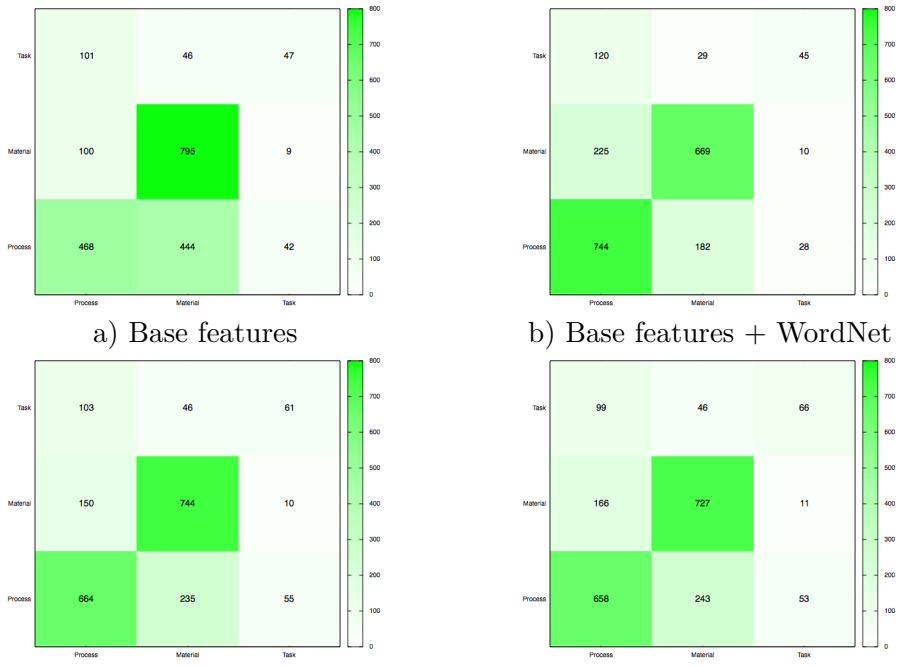
Figure 4.9 *Evaluation of keyphrase extraction using a CRF model trained by type of keyphrase (process, material, task). Choosing the largest keyphrase.*

Bias in types

The confusion matrices, in Figure 4.10, show that TASK is often confused with PROCESS, which in turn seem to be too predominant, indicating a bias in the collection towards this class.

An analysis of the annotated collection showed certain inconsistencies in annotations that may be at the origin of the errors: for instance, in file `2212667814000732.ann`, we found a conflicting annotation for "synthetic assessment method": alone is annotated as PROCESS, but the keyphrase "synthetic assessment method based on cloud theory" is annotated as TASK, which seems odd. In file `S2212671612002351.ann`, we found that "position estimation method" is labelled as TASK, when it should instead be a process.

In Table 4.3 we show the results obtained with different combination of features, compared to the best system at the SemEval 2017 Task 10 in the



a) Base features b) Base features + WordNet

c) Base features + WordNet + embeddings d) Base features + embeddings

Figure 4.10 *Confusion matrices for the 4 configurations tested.*

subtask for *type classification* (Buscaldi et al. 2017).

Table 4.3 *F_1-measure for each test configuration, compared with best keyphrase extraction at SemEval 2017 Task 10.*

	PROCESS	MATERIAL	TASK	all
Base	.577	.726	.322	.619
Base+WN	**.728**	.750	.325	.700
All features	.710	**.778**	.381	**.716**
Base+Embeddings	.701	.764	**.407**	.701
best@SemEval2017	.660	.760	.280	.670

From these results and the confusion matrices in Figure 4.10 it can be seen that WordNet features are helpful in discriminating the MATERIAL from the PROCESS class. The word embeddings features had a positive impact on the TASK class, which is the most difficult to identify.

4.3 Final results

In this section we present the final results obtained using the approach described in Chapter 3. This results were achieved using the package Kleis described in Chapter 6

BIO notation. In this experiment we compare the baseline with the final approach using the Python package Kleis. Using the same features and the same notation. The difference with preliminary experiments is the change in the pre-processing described in Chapter 3. The results are shown in Figure 4.11.

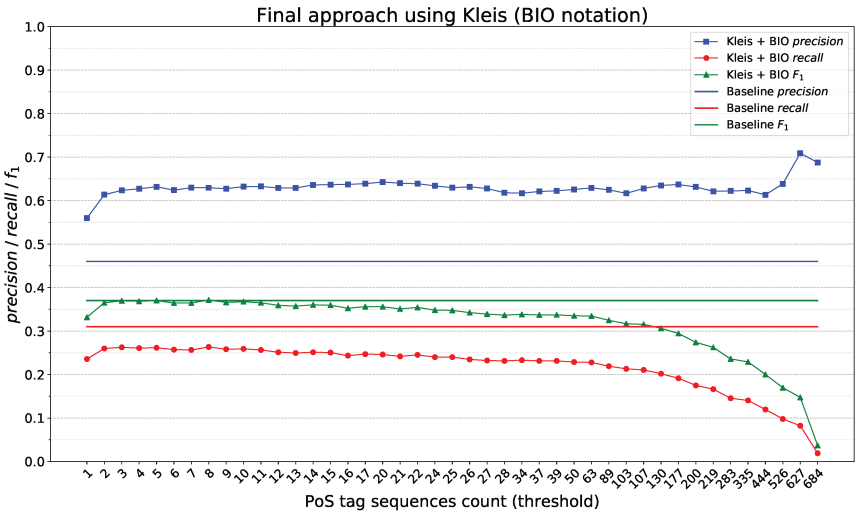


Figure 4.11 *Comparison of the baseline and the final approach (BIO notation).*

BILOU notation. In this experiment we compare the baseline with the final approach, as the previous we used the Python package Kleis. We used the same features, but BILOU notation. The results are shown in Figure 4.12.

PoS tag sequence as a feature In this experiment we compare the baseline with the final approach using the Python package Kleis. In this case we use the BILOU notation. In addition we use the PoS tag sequence as feature. The results are shown in Figure 4.13.

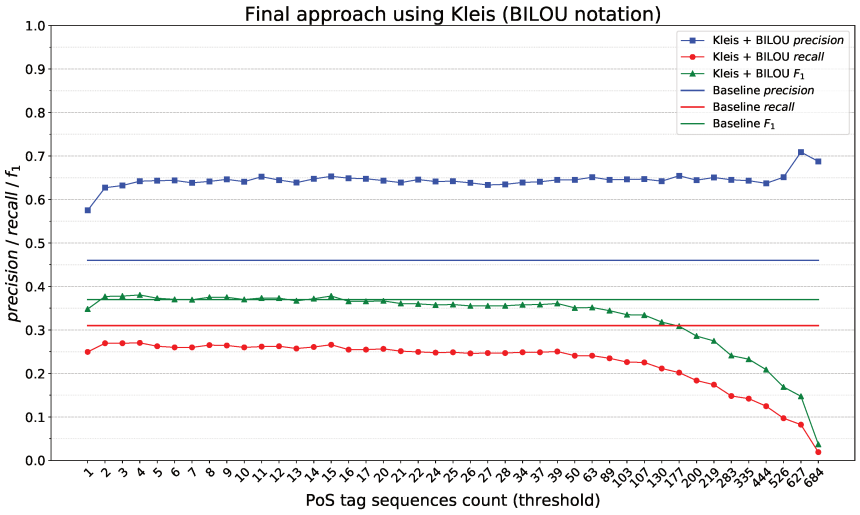


Figure 4.12 *Comparison of the baseline and the final approach (BILOU notation).*

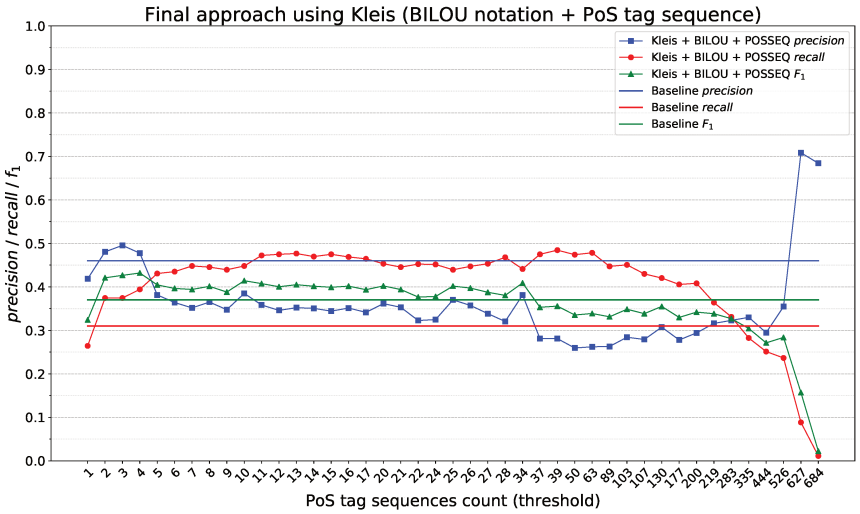


Figure 4.13 *Comparison of the baseline and the final approach (BILOU notation + PoS tag sequence).*

4.4 Summary

We reached the best results of our proposal using a CRF model trained with candidate keyphrases filtered with PoS tag sequences selected by their occurrences in the corpus. We removed the overlapping of identified keyphrases, in a post-processing step, by choosing the largest labeled keyphrase.

Chapter 5

Conclusions and future work

We have addressed the task of automatic keyphrase extraction, with the aim of characterizing scientific documents to improve the search of *state of the arts*. Filtering candidate keyphrases is an important stage for the task and can improve the performance of the extraction method. Using *sequences of part-of-speech tags* (*PoS tag sequences*) is a common method to address the filtering phase. Previous filtering methods focused on *noun phrases* and *prepositional phrases*.

We experimented with the use of PoS tag sequences to filter candidate keyphrases with the assumption that there are syntactic patterns in the tag sequences as consequence of the style used to write keyphrases. We think this notion could help to recognize keyphrases. We filtered candidates using sets of PoS tag sequences selected by their frequency in human annotated corpora. We applied the same principle to test the use of regular expressions based on the PoS tag sequences to filter candidates and compared the results.

We tested SVM and CRFs using filtered candidates with our approach, obtaining improvements in $F_{1.0}$-score by increasing the recall. We have made available lists of PoS tag sequences, the regular expressions and the synsets used in this work. We also released a Python module for keyphrase extraction using the approach here described. It can be found in the Python repositories under the name of *kleis-keyphrase-extraction*, see next chapter for details.

5.1 Conclusions

Our results show that filtering candidate keyphrases with our method improves the *recall* using CRF models and SVM, however, those methods present a lost in *precision* in comparison with our baseline. Our experiments also show that it is possible to outperform the *baseline* in $F_{1.0}$ depending on

the subset of PoS tag sequences used, it is evidence that a wise selection of sequences of part-of-speech tags in the filtering stage improves the results on the task of automatic keyphrase extraction.

We obtained better results using the PoS tag sequences as a feature to train CRFs models with the filtered candidates, i.e., *NN NN*, *NNS NN*. It contrast with other works reporting that there is not an improvement doing this using other methods (e.g., logistic regression) to identify keyphrase. It is possible that it is only true for CRFs, more experiments are needed to confirm that it is true for other methods.

According to our experiments, it is more useful to filter candidates with the PoS tag sequences picked by their occurrence rather than to designing fixed patterns to extract specific types of text segments (e.g., noun phrases or prepositional phrases). At the same time filtering in this way reduces the lost of annotated candidates and reduces the dimensionality. We found that using regular expressions has a negative impact in *recall* without significant improvement in *precision*.

In the task of *"keyphrase type classification"*, that we addressed secondarily, we integrated external knowledge, acquired either from an existing resource like WordNet or learned from a large corpus of text and encoded using word embeddings, as features for a SVM classifier. The obtained results outperform those obtained by the best system presented at SemEval-2017 Task 10. Our method presents margins of improvement, since some parameters were chosen arbitrarily and further investigation is needed to discover the optimal ones. The experiments also highlighted some problems with the SemEval 2017 Task 10 collection: one of the classes seems underrepresented and our analysis exposed a certain number of annotation errors which may require a manual re-annotation.

5.2 Discussion

We showed that using PoS tag sequences to filter candidate keyphrases improves the performance of the task of keyphrase extraction. However, there is not a clear explanation of why it works. Mostly because, as we explained, the considerations to define a keyphrase are relative to the reader (or annotator) of a given text. Those considerations are the *"relevance"* and the *"usefulness"* of the word sequences within a text. Specifically to the *"relevance"*, we think that some concepts in scientific texts are named to denotate (intentional or unintentional) importance in the text based on their writing pattern (e.g. inflexions, made-up words). Thus, part of this relevance in the writing pattern could be latent in the PoS tag sequences. Based on this as-

sumption, we think it is a plausible explanation of why, despite its simplicity, the filtering phase with the selection of PoS tag sequences (ranked by their occurrence) produces interesting improvements with regard to the unfiltered method. In order to maximize the results, a better approach to select the PoS tag sequences is needed.

It is not clear if the negative impact in performance using the regular expressions based on PoS tag sequences is proper of the evaluated corpus (e.g. SemEval 2017), we think that this approach deserves further study and it is recommended to test with other resources.

5.3 Future work

We need to test the performance of the approach here proposed for *automatic keyphrase extraction* with other available resources, (i.e., Inspec corpus). This method could be easily extended to other languages, (e.g. French and Spanish,) to observe if using PoS tag sequences in the filtering stage has a similar behavior to the described in this work.

More experiments are required to get more insight in the matter and determine if filtering candidates with PoS tag sequences have the same effect that we observed using other techniques to identify keyphrases, i.e., logistic regression, ranking methods.

We would like to explore the inclusion of semantic information to observe the effect in the precision on the extraction, however, related works using word embeddings shows that this approach does not improve the performance.

Chapter 6

Kleis - Python package

Kleis is a python package for automatic keyphrase extraction from scientific text. This package uses a CRFs model (Conditional Random Fields) to label keyphrases in text, the model is trained with candidate keyphrases filtered with part-of-speech tag sequences.

It is an implementation of the approach described in the previous chapters. It is currently available in GitHub[1]. It is a development release to use with caution in production environments. The latest release is the version *r0.1.2*[2]. The version *r0.1.3*[3] is under development, it includes models trained with other datasets and an evaluation module. The next version is going to include models trained with a French dataset.

This package was originally developed to test our approach for automatic keyphrase extraction described in this document. Then it was extended to ease the testing of different features, datasets and labeling notation. The first implementation was made to run using the SemEval 2017 Task 10 dataset (Augenstein et al. 2017). It is found in GitHub under the name *kpext*[4]. It is not easy to extract keyphrases using this first version because it is an simple integration of individual scripts. Then it was integrated in a library to help our colleagues to extract keyphrases using our approach. Currently, it has been uploaded to the Python package-management system under the name *kleis-keyphrase-extraction*[5].

[1] https://github.com/sdhdez/kleis-keyphrase-extraction/
[2] https://github.com/sdhdez/kleis-keyphrase-extraction/tree/r0.1.2
[3] https://github.com/sdhdez/kleis-keyphrase-extraction/tree/r0.1.3
[4] https://github.com/sdhdez/test-scripts/tree/master/kpext
[5] https://pypi.org/project/kleis-keyphrase-extraction/

6.1 Description

The latest released version is the *r0.1.2*, it only includes the CRF models trained using the PoS tag sequences from the training dataset provided in SemEval 2017 Task 10[6]. The features used to train the models are described in Subsection 3.3.1 using BIO[7] or BILOU[8] notation.

Table 6.1 *Combination of features and labeling notation of the CRF models distributed in* kleis *version* r0.1.2.

Notation	Features	
	simple	Simple + PoS tag seq.
BIO	✓	×
BILOU	✓	✓

6.1.1 Installation

The easiest way to install the package is using the Python Package Installer (pip[9]). It includes the pre-trained models. The following command installs the latest released version (r0.1.2).

```
$ pip install kleis-keyphrase-extraction
```

To install the latest version or to contribute on the development it is needed to clone the GitHub repository. Notice that the code on GitHub doesn't include the pre-trained models. The training is going to generate a CRF model using the given parameters if it doesn't exists, see how to add the datasets in the following subsection.

```
$ git clone https://github.com/sdhdez/kleis-keyphrase-extraction.git
$ cd kleis-keyphrase-extraction/
```

It is possible to work with the non-released version *r0.1.3* of *kleis*, but first it is needed to change the git branch. The following command does this step in the cloned repository.

[6] https://scienceie.github.io/resources.html
[7] **B**eginning, **I**nside, **O**utside
[8] **B**eginning, **I**nside, **L**ast, **O**utside, **U**nit
[9] https://pip.pypa.io/en/stable/

```
$ git checkout r0.1.3
```

To work directly on the code while editing the code is possible to use the following command inside the directory of the previously cloned repository[10].

```
$ pip install -e .
```

Also, it is possible to generate an installable and distributable package with the following commands in the cloned repository. The version depends on the field version in the file *"setup.py"*

```
$ python setup.py sdist bdist_wheel
$ pip install dist/kleis_keyphrase_extraction-0.1.3.dev0-py3-none-any.whl
```

6.1.2 Datasets

The pre-trained models are based in four datasets, i.e. SemEval 2010 Task 5, RANIS, ACL RD-TEC 2.0 and SemEval 2017 Task 10.

If you are installing from source code, first, it is needed to download the desired dataset in the home path. For example, the SemEval 2017 dataset should look as follows.

```
$ ls -1 ~/kleis_data/corpus/semeval2017-task10
brat_config/
dev/
eval.py*
eval_py27.py*
README.md
README_data.md
README_data_dev.md
scienceie2017_test_unlabelled/
semeval_articles_test/
train2/
util.py*
xml_utils.py*
```

Notice that the first run is going to take a while, since it need to train the model.

[10] https://pip.pypa.io/en/latest/reference/pip_install/#editable-installs

6.2 Usage and examples

The first step to test the package should be running the following command to confirm that it is working.

```
$ python keyphrase-extraction-example.py
```

There are other useful examples in the directory *notebooks*.

6.3 Future development

This in a package in development and should be used with caution in production environments.

Future development includes.

- Completion of the unit tests.
- Addition of pre-trained models.
- Addition of training datasets.
- Addition of labeling methods.
- Addition of options.

Part III

Semantic Similarity

Chapter 7

State of the art on semantic similarity

In this chapter we describe distinct similarity measures and the state of the art on word and document representations commonly used in the measurement of semantic similarity.

It is common for researchers to find difficulties when they are looking for the *state of the art* in their respective fields. We think that the retrieval of documents could be specially challenging if the lexical representation of important concepts in scientific documents are relatively of new usage or simply if they are not widely used. Additionally, looking for widely known terms could not have the expected results if the lexical representation is being used among different fields with different meanings. In some cases, important concepts might have different lexical representations.

One common problem is the use of different lexical representations for the same concept. For example, previously we explained that the terms *"keyword"* and *"keyphrase"* are used as synonyms, or more strictly as equivalent terms to avoid confusion. Both terms allude *to a sequence of words describing the main topics of a document.* In fact, there are other lexical representations used in academic literature on the context of automatic keyphrase extraction, for example, *"key term"*, *"key segment"* or variations like *"key phrase"* and *"key-phrase"*. All those differences in the representation depend on the origin of the terms and the context in which they are being used, e.g., authors, writing date, field of study between others.

Intuitively, one could assume that the lexical content in a pair of documents is related to their semantic similarity. This assumption constitutes (or it's one of the premises of) the *distributional hypothesis* (Firth 1957; Harris 1954). If important concepts addressed in a pair of scientific documents are semantically similar it is likely that both documents share content words. It

should be true even if those concepts are lexically different, because there is a relationship between semantic similarity and lexical *context* (George A. Miller and Charles 1991; Rubenstein and Goodenough 1965). Alternatively, if both documents share content words it is likely that they are addressing similar, or at least related concepts, in some extent.

We consider that *measuring semantic similarity* between documents is a fundamental step in the generation of state of the art. It can be used to improve the retrieval of documents required by researchers.

Applications of document-level semantic similarity are diverse, e.g., document retrieval (Salton and Buckley 1988), news categorization and clustering (Greene and Cunningham 2006; Ontrup and Ritter 2001), song identification (Brochu and de Freitas 2003), and multilingual document matching (Quadrianto et al. 2009).

7.1 Definition of semantic similarity

Comparing concepts extracted from a document is a relatively easy task for humans. Take for example the following pairs of words, *"car"* and *"vehicle"* or *"car"* and *"mountain"*, the lexical elements in both pairs are different. It is not clear how to measure their semantic similarity using those sequences of characters. Still, a person can easily figure out which word pair is semantically more similar. Nevertheless, it is not easy to make a similar decision in the presence of pairs of terms from scientific literature, e.g., *"X-ray absorption spectroscopy'* and *"XAS"*, or *"X-ray absorption spectroscopy'* and *"liquid water"*. Therefore, generating a corpus to evaluate the measure of *semantic similarity* between scientific documents requires specialized knowledge and skills, along with a great effort to analyze and understand those documents.

Harispe et al. (2015) give a general definition of *semantic measures* that can be generalized for the comparison of different types of elements, word-to-word, concept-to-concept, text-to-text. Semantic measures are defined as:

> *mathematical tools used to estimate the strength of the semantic relationship between units of language, concepts or instances, through a (numerical) description obtained according to the comparison of information supporting their meaning.*

George A. Miller and Charles (1991) make a more specific statement about *semantic similarity*, which is defined as a function of word contexts.

> *"Strong Contextual Hypothesis: Two words are semantically similar to the extent that their contextual representations are similar."*

This definition by George A. Miller and Charles (1991) is based on of *distributional semantics*, which is the assumption that the notion of *semantic similarity* can be defined in terms of linguistic distributions (Lenci 2008). It is more known as the *distributional hypothesis* that is defined by Harris (1954) as:

> *"If we consider words or morphemes A and B to be more different in meaning than A and C, then we will often find that the [contextual] distributions of A and B are more different than the [contextual] distributions of A and C. In other words, difference of meaning correlates with difference of [contextual] distribution".*

This is a popular definition of major importance in computer linguistics (Harispe et al. 2015). It is considered the basis of corpus-based measures, often referred as *distributional measures* (Mohammad and Hirst 2012).

Distributional models, based on the distributional hypothesis, are models of word meaning that are grounded in empirical observations and rest on a solid theoretical foundation (Sahlgren 2008). The similarity between two words based on this approach is referred as *distributional similarity* (Kolb 2009; Lenci 2008; Sahlgren 2008; Weeds 2003). Observe that the same principles applied on words can be used in documents (Harispe et al. 2015).

There are authors pointing the disadvantages of using *distributional similarity*, its applications and differences with others measures. For example, the following observation can be considered as a disadvantage of the distributional approach.

> *"Distributional approaches acquire meanings by virtue of being based entirely on noisy, vague, ambiguous and possibly incomplete language data".* (Sahlgren 2008)

When the data changes the model also changes, thus *distributional models* are *context-sensitive* (Lenci 2008; Sahlgren 2008). Weeds (2003) points the difference between *semantic similarity* and *distributional similarity*. Semantic similarity can be thought of as the degree of synonymy of two words in a sentence, while distributional similarity considers only the impact in the contextual coherence after replacing two words. Under this definitions, semantic similarity implies distributional similarity but distributional similarity does not imply semantic similarity. Harispe et al. (2015) go further to analyze different definitions of the notion of semantic similarity used in the literature to make additional distinctions, i.e., *semantic distance*, *semantic dissimilarity* and *taxonomic distance*.

Harispe et al. (2015) states that *semantic similarity* should be distinguished from the concept of *semantic relatedness* (Pedersen et al. 2007; Resnik 1999). Harispe et al. (2015) highlights the differences defining *semantic relatedness* as

> *"the strength of the semantic interactions between two elements with no restrictions on the types of the semantic links considered"*,

and *semantic similarity* as the

> *"subset of the notion of semantic relatedness only considering taxonomic relationships in the evaluation of the semantic interaction between two elements"*.

Unfortunately, taxonomical resources rely on lexical resources which are not available for many languages and have limited coverage, particularly in specialized domains (Kolb 2009). WordNet (George A Miller 1995) is an example of this resources.

We take into consideration this remarks, however we do not have a taxonomic resource of concepts used in different domains of scientific literature. Therefore we use the notion of *distributional similarity* based on *distributional semantics* (Harris 1954; Lenci 2008; Sahlgren 2008).

Scientific documents are written to address specific topics in a specific domain. A scientific document contains semantically related concepts whose meaning depends on the topics covered in the document. The main concepts in a document contribute to, or affect, the meaning of the other concepts in the same document. With this in mind, we consider that the main concepts in a document are the contextual representations of the document itself. Therefore, two documents are semantically similar to the extent that their main concepts are similar. We consider that *measuring semantic similarity* between scientific documents is the estimation of the strength of the semantic relationship between their main concepts.

7.2 Word similarity

Distributional measures are the most studied *corpus-based semantic measures* (Harispe et al. 2015). They rely on *distributional semantic models*, also known as vector spaces, semantic spaces, word spaces, corpus-based semantic models, semantic models, distributional models (Baroni and Lenci 2010; Harispe et al. 2015; Sahlgren 2008).

7.2.1 Word representations

Distributional measures and distributional models are related to spatial representations, or in other words, the semantic space which characterizes a corpus and the words to compare, e.g. Vector Space Models (VSM) and topic models. In this representation words are considered as points of a highly multi-dimensional space to capture the meaning of words (Harispe et al. 2015; P. D. Turney and Pantel 2010).

Distributional measures differ regarding the type of distributional models, e.g geometric/spatial approach, set-based approach and probabilistic approach (Harispe et al. 2015; P. D. Turney and Pantel 2010). There are different general steps to construct a distributional model, e.g. language models, n-gram models, neural network models, compositionality (Harispe et al. 2015).

- Pre-processing of the text (optional). Filtering stop words or using part-of-speech.
- Defining the context. It is the context which is used to characterize a word: it may be a document, a paragraph, a sentence, a window of words or a number of letters.
- Frequency weighting (optional). It is used to add informativeness of contexts to the word frequency, e.g. the most popular technique is *TF-IDF* (Salton and Buckley 1988).

$$tf \cdot idf_{i,d} = tf_{i,d} \cdot idf_i \tag{7.1}$$

with $tf_{i,d}$ the frequency of the term t_i in a document d and idf_i the inverse document frequency of t_i is defined by:

$$idf_i = log \frac{|D|}{|D_{t_i}|} \tag{7.2}$$

with $|D|$ the cardinality of the set of documents in the corpus and $D_{t_i} \subseteq D = \{d_j \in D | t_i \in d_j\}$ is the set of documents in which the term t_i occurs.

- Dimensionality reduction (optional). The idea is to reduce the number of dimensions, it can be done by removing most frequent contexts, matrix factorizations like Singular Value Decomposition (SVD) (Berry et al. 1995) or other types of techniques like Principal Component Analysis, Independent Component Analysis (P. D. Turney and Pantel 2010), etc.

Numerous spatial approaches based on multidimensional representation of words exist, the most common are Latent Semantic Analysis (Deerwester et al. 1990; Thomas K Landauer and S. T. Dumais 1997; Thomas K. Landauer et al. 1998), Explicit Semantic Analysis (Gabrilovich and Markovitch 2007), Hyperspace Analogue to Language (Lund and Burgess 1996), Schütze word space (Schütze 1993), Random indexing (Kanerva et al. 2000) and Correlated Occurrence Analogue to Lexical Semantic (Rohde et al. 2006). Also, there are important probabilistic approaches such as Probabilistic Latent Semantics Analysis (Hofmann 1999) and Latent Dirichlet Allocation (Blei et al. 2003).

Recently, in the spatial approaches, neural network models are commonly used, including their word representations often called *word embeddings* as introduced by Bengio et al. (2006). These are vector representations of words that capture a certain number of syntactic and semantic relationships generated with neural networks, e.g Word2vec (Mikolov, Chen, et al. 2013; Mikolov, Sutskever, et al. 2013), GloVe (Pennington et al. 2014), FastText (Bojanowski et al. 2017) and ELMO (Peters et al. 2018). We use Word2vec (Mikolov, Sutskever, et al. 2013) which is one of the most popular word embeddings.

The most popular spatial approach is *LSA - Latent Semantic Analysis* LSA, also called Latent Semantic Indexing (LSI) in Information Retrieval. This approach is used to represent a word-context matrix, generally a word-document matrix, which can be used to extract a distributional representation of a word. In this approach, the sparseness of the matrix is reduced using Singular Value Decomposition which is a linear algebra operation used to reduce the number of contexts considered in the matrix.

The most popular probabilistic approaches are *PLSA - Probabilistic Latent Semantics Analysis* that is a statistical technique based on mixture decomposition which are derived from latent class model. The latent variables which are considered in PLSA correspond to topics. The probabilistic model relies on the probability that a word is associated to a given topic and the probability that a document refers to a topic. And *LDA - Latent Dirichlet Allocation* similar to PLSA but with the assumption on the topic distribution in document.

7.2.2 Measures

For this work we used the spatial approach which is based on the assumption that compared elements are defined in a semantic space corresponding to the intuitive spatial model of similarity proposed by cognitive sciences (Harispe et al. 2015).

Words are represented through their corresponding vectors in a matrix.

In this approach, a similarity measure on words correspond to a measure on the respective vectors. Two words are compared by their relative location in a multi-dimensional space, in which the dimensions are the semantic space. There is an extensive list of measures from which we described the most common and those that we use in this document.

As notation for the following descriptions, we define $\vec{u}$ and $\vec{v}$ as the vector representations of the words u and v, with n the size of the vectors, and $\vec{u}_k$ the value of $\vec{u}$ in dimension k.

Among the measures commonly used for comparing vectors is the L_2 Euclidian distance which is an instance of the Minkowski L_p distance for $(p = 2)$. The L_1 Manhattan distance $(p = 1)$ is also an instance of this measure.

$$dist_{L_P}(u, v) = \left(\sum_{k=1}^{n} |\vec{u}_k - \vec{v}_k|^p \right)^{\frac{1}{p}} \tag{7.3}$$

We also use the cosine similarity which is one of widely used in different approaches. It is the cosine of the angle between the vectors u and v.

$$sim_{cos}(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{||\vec{u}||||\vec{v}||} \tag{7.4}$$

It is also possible to use correlation measures (Ganesan et al. 2003; Schütze 1998). For instance, the similarity of two words can be defined as the coefficient of the Pearson's correlation between their vector representations. Note that in the following chapter we use this correlation to measure the correlation on similarity measures between documents, not as a word similarity measure as suggested here.

The Pearson's correlation is defined as the covariance of two variables, $cov(X, Y)$, divided by the multiplication of their standard deviations, σ_X, σ_Y.

$$\rho_{X,Y}\left(X_i, Y_i\right) = \frac{cov(X_i, Y_i)}{\sigma_{X_i}\sigma_{Y_i}} \tag{7.5}$$

Where $X_i, Y_i \in \mathbb{R}^n$ are vectors. The possible values are $-1.0 \leq \rho_{X,Y} \leq +1.0$, a value of $+1.0$ means that X and Y are perfectly correlated, zero means that there is no correlation, and -1.0 means that X and Y behave oppositely.

Word measures not based on spatial approaches are not explained in this document since they were not used, e.g. WordNet similarity, set-based measures, e.g., Dice index, Jaccard coefficient (Bollegala et al. 2007), Pointwise Mutual Information (PMI) (Fano 1961), Maximum likelihood Estimate (Dagan et al. 1999), Kullback-Leibler divergence (Cover and Thomas 2006),

Jensen-Shannon divergence (Cover and Thomas 2006; Wartena and Brussee 2008) and Kendall's τ.

7.3 Document similarity measures

Most of the document similarity measures are extensions of, or are based on, measures which have been defined for comparing words, e.g. topic models such as Latent Semantic Analysis (Lintean et al. 2010) or Latent Dirichlet Allocation (Blei et al. 2003).

For analysis and comparison in our work we use spatial approaches that are analogous to the word representation, but instead of words we used documents. We also use set-based approaches, in which documents are compared by their content words and their intersections, using Dice index and the Jaccard coefficient (Bollegala et al. 2007).

7.3.1 Document representations

In this section, we describe the document representations from approaches not used for word representation. In set-based approaches each document d is a binary representation of *bag-of-words*. In the spatial approaches a document can be a *centroid of word embeddings* Brokos et al. (2016). For example, the most known pre-trained embeddings are the word2vec embeddings (Mikolov, Yih, et al. 2013) in the Google News model[1], it contains 3 million words and phrases represented by 300-dimensional vectors. Note that it is not likely that a pre-trained model could include all the possible words of a vocabulary, yet if it is large enough (e.g. 3 million words). It is because of the vocabulary sparsity of the training dataset. Additionally, in scientific literature it is common to find the presence of new words.

Bag-of-words. It is a set of the words within the document, $w \in bow(d)$, where $w \in V$ is a word from the vocabulary V and d is the document. It is usual to remove *stop words* in the document.

Word2vec centroids. It is a vector representation, each document d is represented by the centroid $\vec{q}$. The centroid $\vec{q}$ is the mean of the word embeddings for each word in the document $w \in bow(d)$.

[1] https://code.google.com/archive/p/word2vec/

$$\vec{q} = \frac{\sum_{w \in bow(d)} \vec{w}}{|bow(d)|}$$

Where $\vec{w}$ is the word embedding for w from $bow(d)$.

Word2vec weighted centroids. It is a centroid $\vec{t}$ for document d using word embeddings, but weighted with the word frequency $tf(w, d)$ and its inverse document frequency $idf(w)$ (Manning et al. 2010; Salton and Buckley 1988). This measure is described by Brokos et al. (2016).

$$\vec{t} = \frac{\sum_{w \in bow(d)} \vec{w} \cdot tf(w, d) \cdot idf(w)}{\sum_{w \in bow(d)}}$$

7.3.2 Measures

The measures for spatial approaches from document representations are the same used for word representation, presented in Subsection 7.2.2.

Given that our goal is to contrast the difference between measuring similarity with pure lexical content and information from the semantic space, we used the *Jaccard index* to compare with the cosine similarity, previously defined in equation 7.4.

The similarity measured using the Jaccard index, sim_{bow}, uses the *bag-of-words* for each pair of documents, $(d_i, d_j) \in D \times D$, where D is the set of documents and d_i, d_j are documents, $d_i, d_j \in D$. The Jaccard index, $sim_{bow}(d_i, d_j)$, is the Jaccard similarity for *bag-of-words* for documents $d_i, d_j \in D$, for example,

$$sim_{bow}(d_i, d_j) = \frac{|bow(d_i) \cap bow(d_j)|}{|bow(d_i) \cup bow(d_j)|} \tag{7.6}$$

To measure similarity between document centroids, we used *cosine similarity* given that they are vectors. For both similarity measures (Jaccard and cosine) apply the following definition $0.0 \leq sim(d_i, d_j) \leq 1.0$ and $sim : D \times D \to \mathbb{R}$.

7.4 Available datasets

The Semantic Textual Similarity (STS) datasets are the state-of-the-art resources for comparing measures and systems dedicated to sentence semantic

similarity evaluations[2]. In Table 7.1 there is a short description of the distinct datasets used in the task of semantic textual similarity.

Table 7.1 *Semantic Textual Similarity (STS) datasets*

Dataset	*Dataset description*	*Task goal*
STS 2017, SemEval task 1 (Cer et al. 2017)	Pairs of sentences. Monolingual English, Arabic, Spanish, and cross-lingual English-Arabic, -Spanish and -Turkish	To score the degree of semantic equivalence of sentence pairs.
STS 2016, SemEval task 1 (Agirre, Banea, Cer, et al. 2016)	Pairs of English sentences. Pairs of cross-lingual English-Spanish sentences.	To score the degree of semantic equivalence of sentence pairs.
STS 2015, SemEval task 2 (Agirre, Banea, Cardie, Cer, Diab, González-Agirre, Guo, Lopez-Gazpio, et al. 2015)	Pairs of English and Spanish sentences. Annotation of chunk alignments.	To score the degree of semantic equivalence of sentence pairs. To score similarity/relatedness score between aligned chunks in sentence pairs.
STS 2014, SemEval task 10 (Agirre, Banea, Cardie, Cer, Diab, González-Agirre, Guo, Mihalcea, et al. 2014)	Pairs of English and Spanish sentences.	To score the degree of semantic equivalence of sentence pairs.
STS 2013 (Agirre, Cer, Diab, González-Agirre, and Guo 2013)	Pairs of English sentences. Data for typed similarity task (pilot).	To score the degree of semantic equivalence of sentence pairs. To score the typed similarity.
STS 2012 (Agirre, Cer, Diab, and González-Agirre 2012)	Pairs of English sentences.	To score the degree of semantic equivalence of sentence pairs.

STS 2012. It is a dataset for the pilot task in SemEval 2012[3]. Participants score how similar is a pair of English sentences. Participants also provide a confidence level for each returned result. The sentence pairs have been

[2]Semantic Textual Similarity Wiki http://ixa2.si.ehu.es/stswiki/index.php/Main_Page
[3]STS 2012 https://www.cs.york.ac.uk/semeval-2012/task6/index.html.

manually tagged from 5 (semantic equivalence) to 0 (no relation), e.g, the following two sentences are scored with (5) because they are equivalent,

- The bird is bathing in the sink.
- Birdie is washing itself in the water basin.

two sentences on different topics are scored with (0). The source of the sentence pairs are other publicly available datasets[4], i.e. Microsoft Research Paraphrase Corpus [5], Microsoft Research Video Description Corpus [6], WMT2008 development dataset (Europarl section)[7].

STS 2013. This dataset is for a task designed similarly to the pilot task STS 2012[8]. The training data is the same provided for the STS 2012 dataset and the test data is from related but different datasets. The objective is the same, to score how similar is a pair of English sentences and to provide a confidence score. The sentence pairs have been manually tagged from 5 (semantic equivalence) to 0 (no relation). The test data is from different datasets, i.e. paraphrase sentence pairs, MT evaluation pairs including those from HyTER graphs and GALE HTER data, Gloss pairs.

In addition, the dataset provide tags for the pilot task on typed-similarity between semi-structured records[9]. Participants should compute the similarity between location, author, people involved, time, events or actions, subject, description. The types are scored from 5 (semantic equivalence) to 0 (no relation). The items in this task are taken from Europeana[10]. The official score is based on mean Pearson correlation across all 8 type similarities.

STS 2014, SemEval task 10. It is a dataset for the task 10 at SemEval 2014 on Multilingual Semantic Textual Similarity [11]. It is distributed in two parts, the first part is based on the datasets STS 2012 and STS 2013 with test data from image description, OntoNotes and WordNet sense definition

[4]Detailed information in https://www.cs.york.ac.uk/semeval-2012/task6/data/uploads/datasets/train-readme.txt.

[5]Microsoft Research Paraphrase Corpus (MSR-Paraphrase) http://research.microsoft.com/en-us/downloads/607d14d9-20cd-47e3-85bc-a2f65cd28042/.

[6]Microsoft Research Video Description Corpus (MSR-Video) http://research.microsoft.com/en-us/downloads/38cf15fd-b8df-477e-a4e4-a4680caa75af/.

[7]SMTeuroparl http://www.statmt.org/wmt08/shared-evaluation-task.html.

[8]STS 2013 http://ixa2.si.ehu.es/sts/index.php%3Foption=com_content&view=article&id=47&Itemid=54.html.

[9]STS 2013 typed-similarity http://ixa2.si.ehu.es/sts/data/trial-typed-readme.txt.

[10]Europeana http://www.europeana.eu/.

[11]STS 2014 http://alt.qcri.org/semeval2014/task10/

mappings, news title and tweet comments, deft discussion forum and news and news headlines.

The second part is to enable the evaluation of semantic textual similarity systems for Spanish. The scale differs from the English dataset, from 0 (no relation) to 4 (semantic equivalence). A development dataset of 65 annotated sentence pairs is provided. The test data includes two datasets, one of 324 sentence pairs, and another one of of 480 sentence pairs. No training data is provided.

STS 2015, SemEval task 10. This dataset for semantic textual similarity is similar to the previous tasks[12]. Given two sentences of text, s1 and s2, participants should compute how similar s1 and s2 are, returning a similarity score, and an optional confidence score. It uses the score scale from 0 (no relation) to 5 (semantic equivalence) for the English part and the scale from 0 (no relation) to 4 (semantic equivalence) for Spanish. The main difference in both parts is the source of the test data.

This dataset includes data for the pilot subtask on interpretable STS. Given a pair of sentences, participants need to identify the chunks in each sentence, and then, align the corresponding chunks. For each alignment should be specified three things. A similarity/relatedness score from 5 (maximum similarity/relatedness) to 0 (no relation at all). A type of the alignment, i.e.

- EQUI: both chunks are semantically equivalent in the context.
- OPPO: the meanings of the chunks are in opposition to each other in the context.
- SPE1 and SPE2: both chunks have similar meanings, but chunk in sentence1 is more specific than chunk in sentence2; and, vice versa.
- SIM: similar meanings, but no EQUI, OPPO, SPE1, or SPE2.
- REL: related meanings. but no SIM, EQUI, OPPO, SPE1, or SPE2.
- ALIC: this chunk has not any corresponding chunk in the other sentence because of the 1:1 alignment restriction, but otherwise the chunk would be aligned to some other chunk.
- NOALI: this chunk has no corresponding chunk in the other sentence.

And an optional tags for alignments showing factuality (FACT) or polarity (POL) phenomena.

[12]STS 2015 http://alt.qcri.org/semeval2015/task2/

STS 2016, SemEval task 1. It is dataset provided for the SemEval 2016 task 1[13]. The objective is to test unified frameworks for semantic processing and evaluation. The definition of the task is similar to previous STS tasks in SemEval. This dataset includes two parts, STS Core and Cross-lingual STS. The STS Core is the traditional task, it is paired monolingual sentences drawn from English data sources, evaluated with English sentence pairs on Plagiarism Detection, Q&A Question-Question, Q&A Answer-Answer, Post-Edited Machine Translations and Headlines. Cross-lingual STS involves assessing paired English and Spanish sentences evaluated with Spanish-English bilingual sentence pairs on Plagiarism Detection, Q&A Answer-Answer, Post-Edited Machine Translations and Headline.

STS 2017, SemEval task 1. It is a dataset provided for the SemEval 2017 task 1[14]. The definition of the task is similar to previous STS tasks in SemEval. It includes data for the evaluation of monolingual and cross-lingual sentence pairs, i.e. Arabic-English, Spanish-English, Arabic-Arabic, English-English and Spanish-Spanish.

[13]STS 2016 http://alt.qcri.org/semeval2016/task1/
[14]STS 2017 http://alt.qcri.org/semeval2017/task1/

Chapter 8

Semantic similarity on scientific documents

As we established in the previous chapter, the lexical content in a pair of documents is related to their semantic similarity based on the *distributional hypothesis*. If two scientific documents share content then they could be addressing similar or related concepts. Even if the main concepts were lexically different, they could be similar in some degree, simply because they share common contextual words (Harris 1954; George A. Miller and Charles 1991; Rubenstein and Goodenough 1965; Sahlgren 2008).

Scientific publications are not an exception, it often occurs that abstract concepts have distinct lexical representations or terms; the reasons could vary depending on the origin or the context in which the terms are being used, however, it is hard to identify this change of representation and requires a deeper understanding and specialization. It is constantly a difficulty for researchers while they are looking for the *state of the art* in their respective fields. Being able to evaluate the retrieval of documents in the presence of these phenomena is an essential step to improve the search of the *state of the art* for the given topic.

In this chapter we explain the motivation and challenges of our work on the semantic similarity of abstracts from scientific papers. We explain how we analyzed the correlation of document similarities based on their lexical content and semantic information from *word2vec embeddings*. We compare the correlations between the measures using *bag of words* and extracted keyphrases under the assumption that keyphrases provide semantic information to the measures. We end the chapter with the description of our work towards the generation of a dataset of scientific documents to measure semantic similarity in the presence of abstract concepts with different lexical representations or terms with multiple meanings.

8.1 Motivation

In the first part of this work on automatic keyphrase extraction, we defined keywords or keyphrases as the most relevant sequences of words that better describe the content of a document (Frank et al. 1999; Haddoud et al. 2015; Hammouda et al. 2005; Kim, Medelyan, et al. 2010; Peter D Turney 2000). The *relevance* of a keyphrase is directly related to its *usefulness* for the reader (Frank et al. 1999; Kim, Medelyan, et al. 2010) and the semantic relation with the central topics in the document (Kim, Medelyan, et al. 2010; Peter D Turney 2000).

By definition *keyphrases* include semantic information given their relation with the main topics of a document (Hasan and Ng 2014; Kim, Medelyan, et al. 2010; Peter D Turney 2000; Witten et al. 1999). This relation has been taken for granted, however to this moment we have not found a good analysis of the effect of using keyphrases to measure semantic similarity on scientific documents in comparison to using only content words and *corpus-based semantic measures*.

Our first premise is that pairs of scientific documents can be hard to characterize semantically if they contain relative common vocabulary, even though they are semantically similar. Our second premise is that there are pairs of scientific documents semantically similar with relative high co-occurrence of common technical words and low co-occurrence of important terms, meaning that measures based on lexical co-occurrence could be losing semantic information. For example, consider a document pair containing different important terms, the first contains *"software development system"* and *"cerebral aneurism"*; the second contains *"spatial decision support system"* and *"mean value analysis"*. Those terms help to better characterize the document pair than the following common words in scientific literature *"algorithm"*, *"solution"*, *"system"* and *"problem"*. Third, the assumption is that including semantic information from concepts with different lexical representations improve the retrieval of scientific documents.

To test the validity of the previous premises, we want to analyze the correlation of measuring semantic similarity on pairs of scientific documents using semantic information from the keyphrases and their lexical content.

We think that in the search of the *state of the art* it is hard to retrieve useful documents in the presence of equivalent lexical variations of concepts along the literature (synonymy). Actually, we do not know if this problem affects significantly the retrieval of the *state of the art*. We need evidence of the liability of addressing this problem specifically. In this work, we started by looking for pairs of documents to analyze the correlation of their semantic similarity and lexical content. Identifying those documents can help us to

form a corpus of scientific documents to test measures of semantic similarity emphasizing semantic the use of keyphrases instead of lexical content. It can allow us confirm the assumption that keyphrases are better than the raw lexical content to retrieve scientific documents in the presence of phenomena like polysemy and synonymy.

8.2 Challenges

An example of the challenges to construct specialized corpora is the concept of *"entropy"*, which is widely used among different areas. In information theory the term's name was inspired in the concept used in statistical mechanics because of the similarity of the Shannon's formula. But it doesn't mean that this term is equivalent in a field like information theory, where researchers might not be interested in the term as it is used in the context of classic thermodynamics. Both interpretations of the term, in different fields, share part of their meaning because of the link and inspiration on the abstract concepts, even though we can not say that they are the same. A similar example occurs with the terms *"logistic model"* and *"logit model"*, the second term was introduced as analogy to other concept, *"probit model"* (Cramer 2002). Actually, those terms address to the same concept, they are used depending on the domain of study, however both can be replaced one by another without changing the meaning of a sentence.

Addressing this type of problems to retrieve the *state of the art* along the scientific literature requires the use of semantic information. It has the potential to filter relevant documents by recognizing conceptual mismatches (Naik et al. 2015).

There are datasets to evaluate keyphrase extraction (Augenstein et al. 2017; Kim, Medelyan, et al. 2010; QasemiZadeh and Schumann 2016) or document clustering, e.g. medical publications, news articles, newsfeeds, webpages, emails (Naik et al. 2015). SemEval 2017 Task 10[1] (Augenstein et al. 2017) provided a dataset for keyphrase extraction that includes annotated relationships of synonymy and hyponymy between keyphrases in texts of four domains (i.e. Computer Science, Material Sciences and Physics). However, the relationships are restricted to keyphrases within each document, it does not include relations of keyphrases between different documents. Additionally, there is a small number of annotated relationships and not all the documents in the dataset include them.

As far as we know, there is not yet an available dataset of scientific documents in multiple fields to evaluate the measurement of semantic simi-

[1]SemEval 2017 Task 10 https://scienceie.github.io

larity between them using keyphrases. A corpus for this task should include annotations of synonymy and polysemy between the main concepts of the documents. It should allow the evaluation of document similarity among different domains using main concepts (or keyphrases) in the presence of different phenomena. Constructing a dataset of scientific documents with the required characteristics is challenging. Mostly because the degree of specialization needed to annotate keyphrases QasemiZadeh and Schumann (2016) and to relate them among different fields of research.

Using the most important terms in a document or keyphrases can provide semantic information. However, as we already explained, it is hard to extract the most relevant phrases from a scientific document. Assuming that we achieved to extract the most meaningful keyphrases to characterize a document, it does not solve the problem of synonymy and polysemy. In fact, to this moment we do not know for sure if trying to solve those problems using keyphrases could have a significative impact on scientific document retrieval. Therefore, to corroborate this assumption, first, it is needed to construct a corpus with the given requirements.

8.3 Analyzing correlations of document similarity measures

In this section we describe our analysis of the correlation of document similarities based on their lexical content. We compared the document similarities of papers' abstracts, using the binary co-occurrence of the content words (Jaccard similarly) and their similarity using semantic information from word embeddings (Cosine similarity). We included the correlation of the same measures of document similarities using extracted keyphrases from the abstracts instead of their content words.

Given that keyphrases are the most relevant word sequences related to the main topics of a document, we make the assumption that keyphrases convey the semantic information contained in the document. If it is true that keyphrases are a source of semantic information enough to retrieve the state of the art in scientific literature, instead of using word-based techniques, we could justify deeper studies in the matter. In our work we try to compare the difference between both sources (i.e. keyphrases and bag-of-words) by measuring and analyzing their correlation.

8.3.1 Methodology

We measured similarity between pairs of documents from a sample of 10,000 abstracts from the ACM V9 dataset in ArnetMiner. We used *bag-of-words* and *keyphrases* for comparison, then we empirically analyzed the Pearson's correlation of their similarity measures. Additionally, we compared the correlation of the similarities using semantic information from word embeddings.

Dataset and sampling

We used titles and abstracts publicly available in ArnetMiner[2], specifically the ACM V9 dataset from the Citation Network (Tang et al. 2008)[3], containing 2,385,022 papers (including titles, abstracts, year, publication venue and authors) and 9,671,893 citations. For this work, we randomly selected $10,000$ documents from the ACM V9 dataset, including titles and abstracts without the relations of the citation network, however, in future work, this information could help us to construct a corpus. We restricted the document selection to those containing no less than 5 keyphrases extracted using the Python package *Kleis*[4] from our previous work (Hernandez et al. 2017a) (see Chapter 6). The minimum number of keyphrases was chosen arbitrarily avoiding selection of short abstracts, it was needed because the dataset contains instances in which the abstract is a short phrase or the title of the article. In this way we ensured that the abstracts significant text content for our analysis.

Document representation

The set D is the sample of 10,000 documents, where each document $d \in D$ can be represented by a set of *bag-of-words* or *bag-of-keyphrases* or by a *centroid of word embeddings*, weighted or not.

Bag-of-words $bow(d)$**.** It is a set-based approach, where each document is represented as a set *bag-of-words*. For every document $d \in D$ there is a set of words from the title and abstract of the text, $w \in bow(d)$, where $w \in V$ each word w belongs to the vocabulary V and it is present in the text from document d. We removed *stop words* in the documents using NLTK 3.4[5].

[2]AMiner Dataset https://aminer.org/data
[3]Citation Network in the AMiner Datasethttps://aminer.org/citation
[4]Kleis v0.1.2.dev0 https://github.com/sdhdez/kleis-keyphrase-extraction/releases/tag/v0.1.2.dev0
[5]Official site for the NLTK package https://www.nltk.org/

Bag-of-keyphrases $bok(d)$**.** It is a set-based approach, where each document is represented as a set *bag-of-keyphrases*. For every document $d \in D$ there is a set of *keyphrases* extracted from the title and abstract of the text, $k \in bok(d)$, where k is a keyphrase extracted from the document using *Kleis*[6].

Keyphrases can be multi-word terms in the document, in other words, the keyphrase k is a sequence of words in the vocabulary of keyphrases $k \in K$, where $K = \{(w_1, w_2, w_3, \ldots, w_p) \in V^p\}$ and $p \in \mathbb{N}^+$ is the size of the largest keyphrase in a corpus. For practical reasons, in this representation we consider a keyphrase as an unique entity and not as composition of words.

Word2vec centroid $\vec{q}$ **for words.** It is a spatial approach, where each document is represented as a *word2vec centroid*. For every document $d \in D$ there is a vector $\vec{q}$, which is the mean of the word embeddings for each word in the document, $w \in bow(d)$. De Boom et al. (2016) showed that it's possible to exploit the properties of embeddings to represent sentences with the average, the max, or the min of the vectors of the composing words. We implemented the method described by Brokos et al. (2016) to obtain the centroids, which is

$$\vec{q} = \frac{\sum\limits_{w \in bow(d)} \vec{w}}{|bow(d)|}$$

Word2vec centroid $\vec{q}$ **for keyphrases.** It is a spatial approach, based on the previously described *word2vec centroid* $\vec{q}$. The difference is in the use of the tokens of the keyphrases instead of words. Since keyphrases can be multi-word terms they are very sparse and likely they are not in the Google News model of word embeddings. To go around this problem we tokenized the multi-word keyphrases and used the corresponding embeddings to obtain the centroid.

Then, for every document $d \in D$ there is a vector $\vec{q}$, which is the mean of the word embeddings for each token w in the keyphrases extracted from the text of the document, $k \in bok(d)$. The tokens from the keyphrases are $\{w : w \in k, k \in bok(d), k \in K\}$, given that $K = \{(w_1, w_2, w_3, \ldots, w_p) \in V^p\}$ where p is the number of words of a keyphrase.

$$\vec{q} = \frac{\sum\limits_{w \in k, k \in bok(d)} \vec{w}}{\sum\limits_{w \in k, k \in bok(d)} 1}$$

[6]Kleis v0.1.2.dev0 https://github.com/sdhdez/kleis-keyphrase-extraction/releases/tag/v0.1.2.dev0

Word2vec weighted centroid $\vec{t}$ for words. It is a spatial approach based on the *word2vec centroid $\vec{q}$ for words*. It follows the same principle, but the vector for every word w in the document d is also weighted with its TF-IDF, $tfidf_{w,d}$ (Brokos et al. 2016).

$$\vec{t} = \frac{\sum_{w \in bow(d)} \vec{w} \cdot tfidf_{w,d}}{\sum_{w \in bow(d)}}$$

Word2vec weighted centroid $\vec{t}$ for keyphrases. It is a spatial approach based on the previously described *word2vec weighted centroid $\vec{t}$*, but adapted for keyphrases like the *word2vec centroid $\vec{q}$ for keyphrases*. We tokenized the multi-word keyphrases to obtain the centroid from the corresponding word embeddings.

$$\vec{t} = \frac{\sum_{w \in k, k \in bok(d)} \vec{w} \cdot tfidf_{w,d}}{\sum_{w \in k, k \in bok(d)} 1}$$

Document pair's similarity

Each document pair is represented as $(d_i, d_j) \in D_a \times D_b$, where D_a, D_b are subsets of D, $D_a, D_b \subseteq D$. Both subsets have the same number of elements $|D_a| = |D_b|$ and they do not share documents $|D_a| \cap |D_b| = \emptyset$. Thus, each document $d_i \in D_a$ and $d_j \in D_b$ is an element of the sample D, $d_i, d_j \in D$. The similarity between a pair of documents (d_i, d_j) is represented by $sim(d_i, d_j) \in \mathbb{R}$, where $0.0 \leq sim(d_i, d_j) \leq 1.0$ and $sim : D_a \times D_b \to \mathbb{R}$.

Jaccard similarity. Given that our goal is to contrast the measurement of similarity between keyphrases (supposedly providing semantic information) and raw lexical content (*bag-of-words*), we used the *Jaccard index* as base of comparison, because it depends on the sharing words on a document pair.

The Jaccard similarity using *bag-of-words* of a document pair, $(d_i, d_j) \in D_a \times D_b$, is represented by $sim_{bow}(d_i, d_j)$, in other words,

$$sim_{bow}(d_i, d_j) = \frac{|bow(d_i) \cap bow(d_j)|}{|bow(d_i) \cup bow(d_j)|}$$

Correspondingly, the Jaccard similarity using *bag-of-keyphrases* of a document pair, $(d_i, d_j) \in D_a \times D_b$, is represented by $sim_{bok}(d_i, d_j)$, in other

words,

$$sim_{bok}(d_i, d_j) = \frac{|bok(d_i) \cap bok(d_j)|}{|bok(d_i) \cup bok(d_j)|}$$

Cosine similarity. It is used to measure similarity between the centroids of word embeddings. The similarity of a pair of documents represented by *word2vec centroids* is $sim_q(d_i, d_j)$.

$$sim_q(d_i, d_j) = \frac{\vec{q}_{d_i} \cdot \vec{q}_{d_j}}{||\vec{q}_{d_i}||||\vec{q}_{d_j}||}$$

This measure is used for the *word2vec centroids* based on *bag-of-words* and *bag-of-keyphrases*, both are referred by sim_{qw} and sim_{qk}, respectively. The same measure is applied to the similarity on *word2vec weighted centroids*, sim_t. Where sim_{tw} is the similarity of centroids based on *bag-of-words* and sim_{tk} is for *bag-of-keyphrases*.

Correlation

We used the Pearson correlation coefficient to get the correlation between different similarity measures. The objective is to observe how the correlation behaves by document. The Pearson's correlation, ρ, is based on the covariance of two variables, $cov(X, Y)$, and their standard deviations, σ_X, σ_Y, see Equation 7.5.

We measured the correlation of document pair's similarities, i.e. d_i and all its pairs $d_j \in D_b$. The variables in the correlation, X or Y, correspond to those similarities using two different measures, e.g. $m1$ and $m2$,

$$X_{m1,i} = sim_{m1}(d_i, d_j) \quad for \quad all \quad d_j \in D_b$$

$$Y_{m2,i} = sim_{m2}(d_i, d_j) \quad for \quad all \quad d_j \in D_b$$

Where the variables X_i and Y_i are vectors of dimension $n = |D_b|$ and $X_i, Y_i \in \mathbb{R}^n$.

The Pearson's correlation, ρ, for two different similarity measures is defined as follows.

$$\rho\left(X_{m1,i}, Y_{m2,i}\right) = \frac{cov(X_{m1,i}, Y_{m2,i})}{\sigma_{X_{m1,i}} \sigma_{Y_{m2,i}}}$$

After obtaining the correlations of the document similarities we empirically analyzed the results.

8.4 Experimental results

In this section we describe the experiments that we made to analyze the correlation of document similarities based on their lexical content using the binary co-occurrence of the content words and extracted keyphrases (Jaccard similarity) and using semantic information from word embeddings (Cosine similarity).

This section starts with an explanation of how we arranged the similarities to simplify the visualization of the correlation. After, in the next subsection, it continues showing the resulting correlations between the similarity measures.

8.4.1 Visualizing correlations

We ordered correlations of similarity measures to ease their interpretation. For each document $d_i \in D_a$, we obtained its mean of similarity measures with all its pairs in the other set, $d_j \in D_b$. We used these means of measures to sort the documents in descending order.

$$mean_{d_i \in D_a} = \frac{\sum_{d_j \in D_b} sim(d_i, d_j)}{|D_b|}$$

The arrangement helps to visualize the correlations of document similarity measures. For example, Figure 8.1 shows the means of Jaccard similarities measures using *bag-of-words* ($sim_{bow}(d_i, d_j)$) and *bag-of-keyphrases* ($sim_{bok}(d_i, d_j)$). The documents, $d_i \in D_a$, are arranged in descending order by the means of measures using $sim_{bow}(d_i, d_j)$.

In Figure 8.2 are shown the most similar pairs of documents. Each point represents a document pair, where the document d_j is the most similar to a given document d_i. The similarities are measured with the Jaccard index using *bag-of-words* (Left) and *bag-of-keyphrases* (Right). The document pairs in the figures are arranged in both axes by the Jaccard similarities using *bag-of-words* (sim_{bow}) similarly to the previous example, see Figure 8.1. It explains the concentration of points in the top-left corner of the first image and the dispersion in the second.

The difference is clearer in Figure 8.3 which shows different similarity measures from all the document pairs, $(d_i, d_j) \in D_a \times D_b$. All the measures are arranged in both axis by the means of similarity of sim_{bow}.

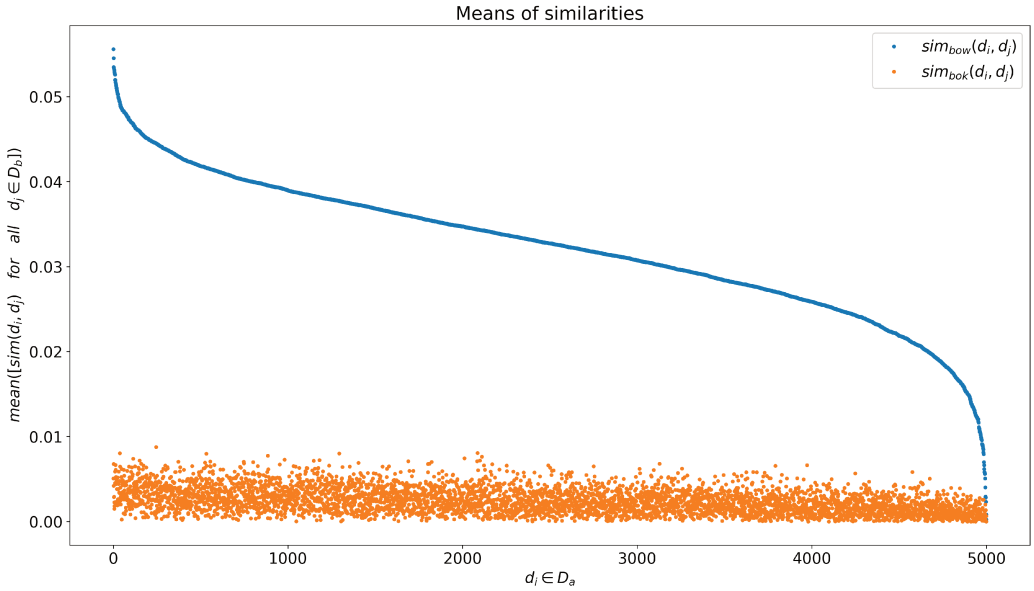


Figure 8.1 *Means of similarities for $sim_{bow}(d_i, d_j)$ and $sim_{bok}(d_i, d_j)$. Arranged by means of similarities of sim_{bow}.*

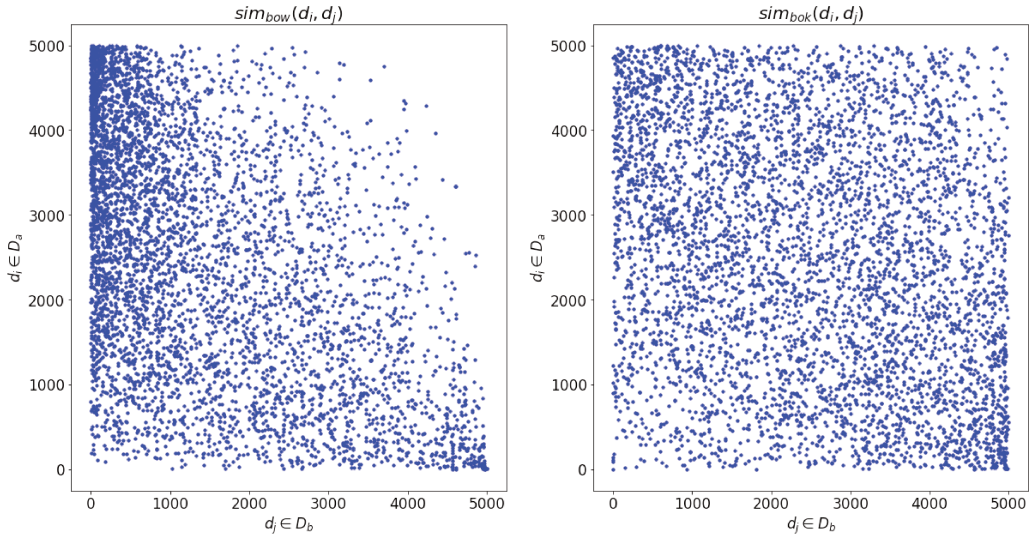


Figure 8.2 *Most similar pairs of documents from $sim_{bow}(d_i, d_j)$ (Left) and $sim_{bok}(d_i, d_j)$ (Right). Arranged by means of similarities of sim_{bow}.*

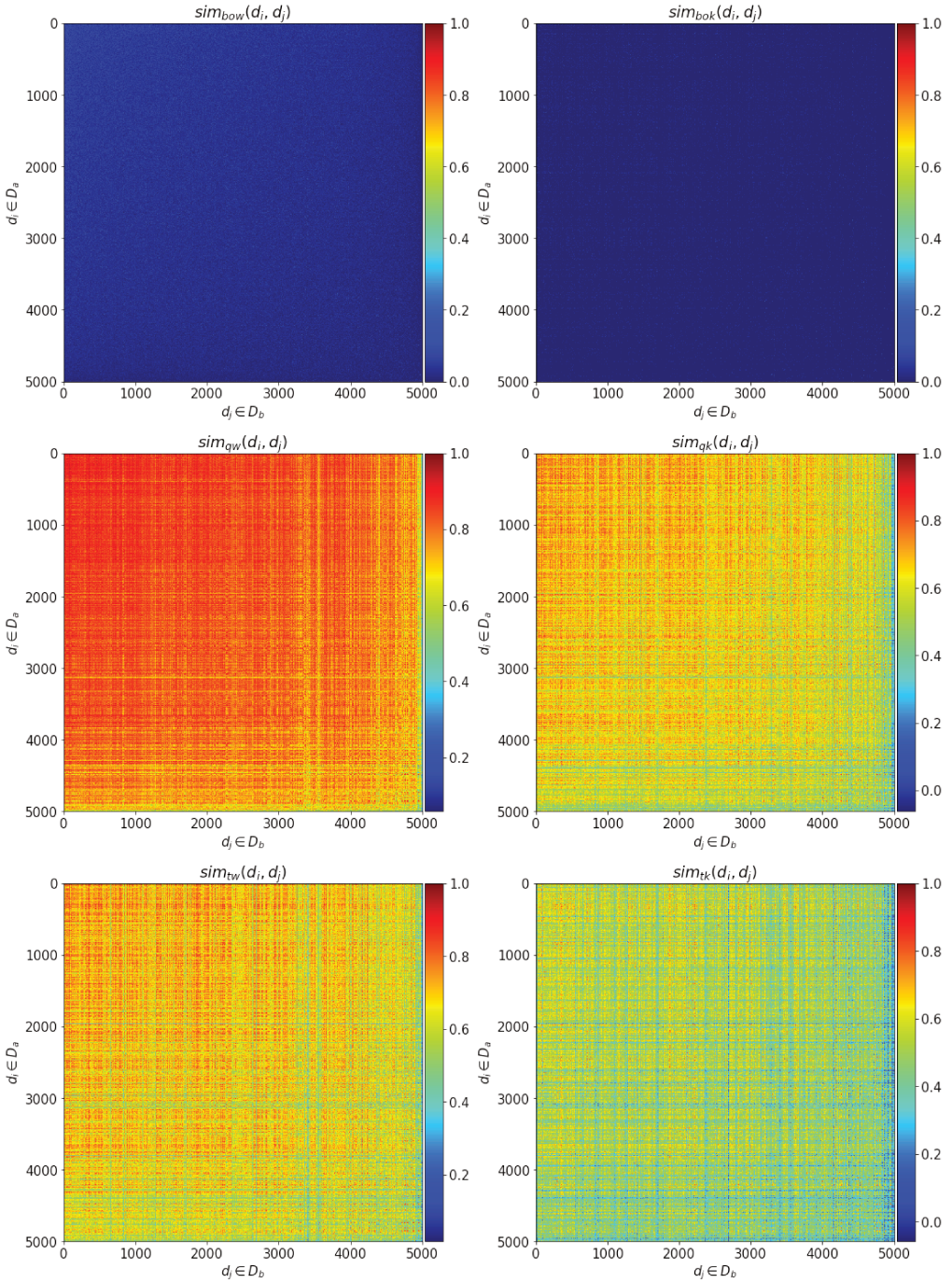


Figure 8.3 *Document similarities using different measures.* $sim_{bow}(d_i, d_j)$, $sim_{bok}(d_i, d_j)$, $sim_{qw}(d_i, d_j)$, $sim_{qk}(d_i, d_j)$, $sim_{tw}(d_i, d_j)$, $sim_{tk}(d_i, d_j)$. *Arranged by means of* sim_{bow}.

8.4.2 Correlations between document similarity measures

As we described before, we measured the Pearson's correlation between similarity measures for each $d_i \in D_a$. In Figure 8.4 is shown the correlation of the Jaccard similarities using and the other measures using *bag-of-words*. The first plot in the top-left corner is the correlation between sim_{bow} and itself, giving a correlation of +1 for all the values.

To contrast the correlations, in Figure 8.5 is shown the correlation of the same similarity measures, though the measure used as point of comparison changes to the *word2vec weighted centroids* using *bag-of-keyphrases*. Thus the similarity means are arranged by sim_{tk} and not by sim_{bow}, as in the previous example. In the same figure, the last plot in the bottom-right corner is shown the correlation between sim_{tk} and itself, resulting in an uniform correlation of +1.

To the left of both figures, 8.4 and 8.5, there are shown the correlations of similarity measures using *bag-of-words*. To the right there are correlations of the measures using *bag-of-keyphrases*. Note that all the correlations are arranged by means of sim_{bow}, as in the previous figures.

To the left of each plot, 8.4 and 8.5, can be found the most similar documents $d_i \in D_a$ to all its pairs D_b, because it the mean is greater than those documents to the right.

Analysis of the correlations

Correlations using *bag-of-words*. The correlations in Figure 8.4 are arranged by means of sim_{bow}. Thus, all documents, $d_i \in D_a$, to the left of the plots contain a common vocabulary since they are in average more similar to their pairs than those documents to the right. These documents on the left share terms with their pairs in a high degree in relation with the size of the document, considering the how the Jaccard similarity is measured. Likely, those terms do not help to improve document retrieval, for example, the following are the five most common terms in the documents in sample D, 'based', 'paper', 'which', 'using', and 'results'.

Observe that sim_{bow} makes its best correlation with sim_{qw}, meaning that *word2vec centroids* are strongly correlated with the *bag-of-words* in each document, mainly because it doesn't included weighting or exclusion of common words. In the other hand, its worst correlation is with sim_{bok}, in which can be observed a few outliers to the right of the plot. We observed that those outliers are documents with one-word-terms as keyphrases, explaining the behavior.

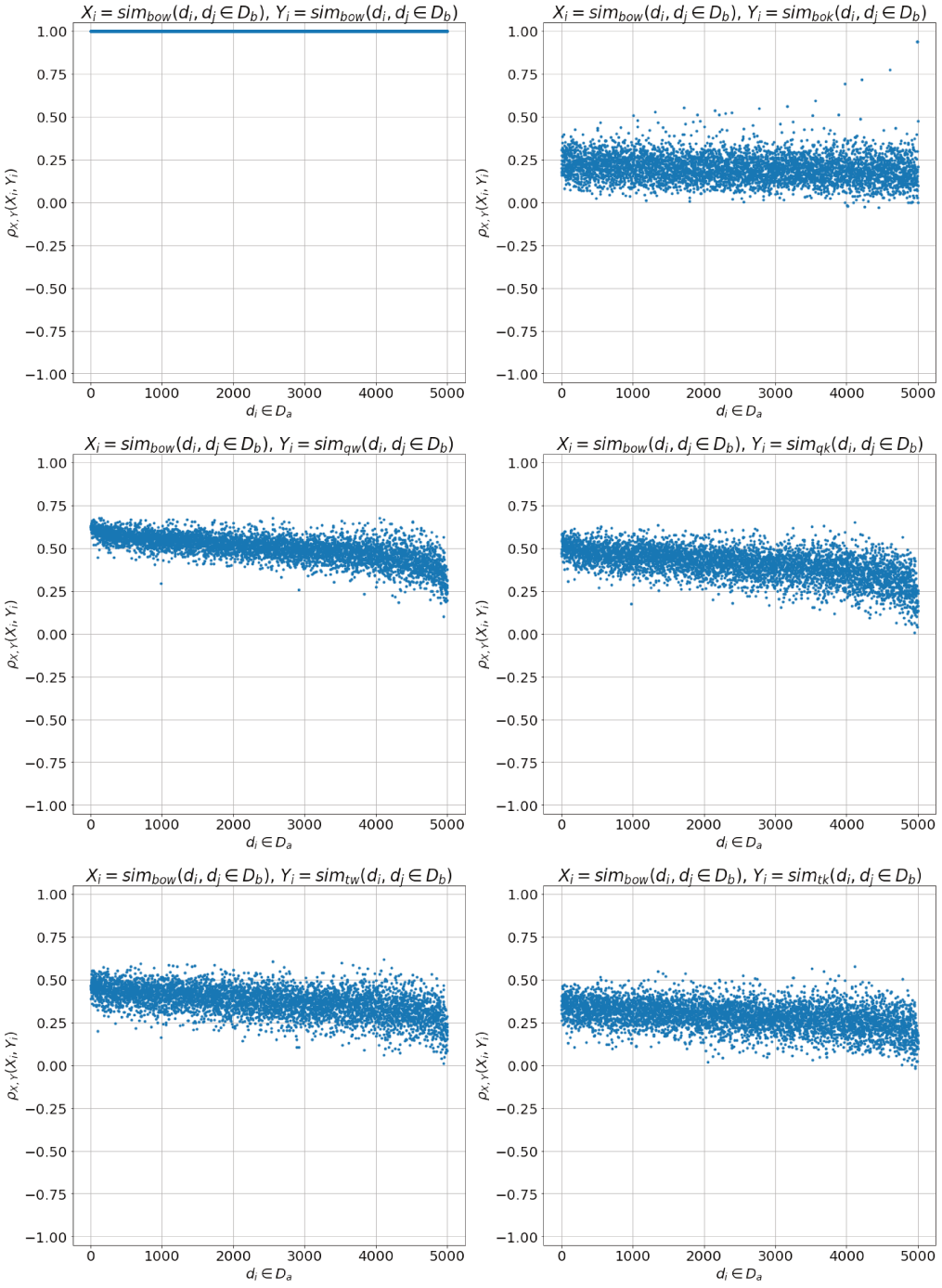


Figure 8.4 *Pearson's correlation of sim_{bow} with different similarity measures for each $d_i \in D_a$ and all its pairs $d_j \in D_b$. The documents $d_i \in D_a$ are arranged by means of sim_{bow}.*

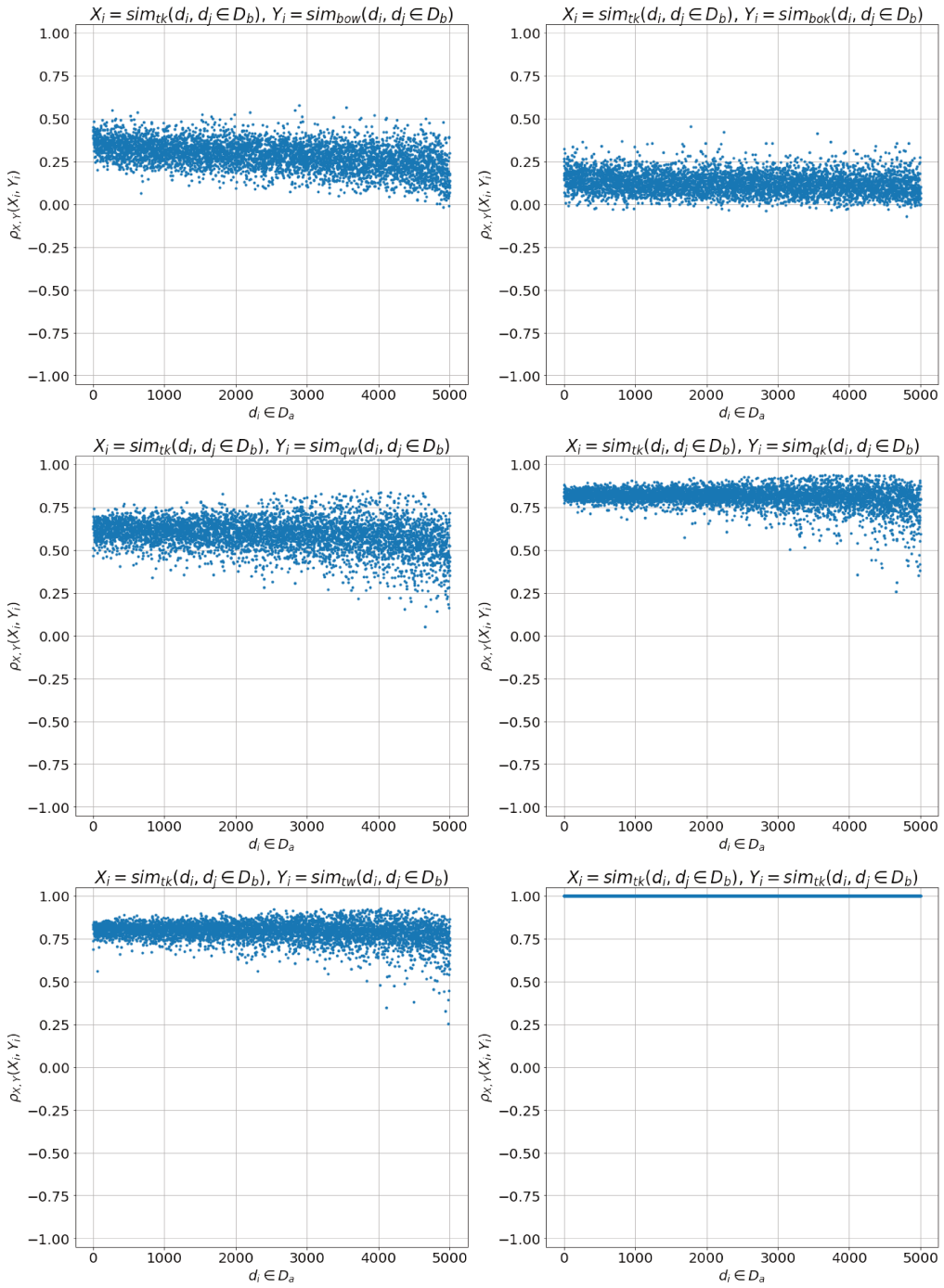


Figure 8.5 *Pearson's correlation of sim_{tk} with different similarity measures for each $d_i \in D_a$ and all its pairs $d_j \in D_b$. The documents $d_i \in D_a$ are arranged by means of sim_{tk}.*

Additionally, note that sim_{bow} is not well correlated with sim_{bok} as it is with the rest of the measures. The corresponding correlations of *word2vec centroids*, sim_{qk} and sim_{tk}, behave similarly. We are getting correlated results using the *word embedding centroids* from keyphrases and the Jaccard similarities using *bag-of-words*. It means that we are obtaining a dimensionality reduction in terms of the used vocabulary. However, it is possible that centroids are as general as *bag-of-words* and we should discuss their effectiveness.

Correlations using *bag-of-keyphrases*. Figure 8.5 is ordered by means of sim_{tk}, contrasting the difference between using *bag-of-words* and *bag-of-keyphrases*. The assumption is that keyphrases provide semantic information in combination with the word embeddings.

On the contrary of what we expected, sim_{bok} and sim_{tk} are not well correlated. In fact, they are worse correlated than sim_{tk} and sim_{bow}. We do not now if is a consequence of using pre-trained word vectors instead of properly trained keyphrase embeddings. It is also possible that using centroids of word embeddings is giving a general representation of a document and not an useful characterization.

Observe that in both figures, documents with a common vocabulary have a worst correlation when measuring document similarities.

8.5 Towards a dataset of scientific documents for semantic similarity

As we described at the beginning of this chapter, as far as we know, there is not an available dataset of scientific documents in multiple fields to evaluate the measurement of semantic similarity using keyphrases. We do know other datasets, to evaluate keyphrase extraction (Augenstein et al. 2017; Kim, Medelyan, et al. 2010; QasemiZadeh and Schumann 2016) or document clustering (Naik et al. 2015).

It is difficult to construct a dataset of scientific documents with the degree of specialization needed. It is hard to extract the most relevant phrases from a scientific document. Assuming that we achieved to extract the most meaningful keyphrases to characterize a document, it does not solve the problem of synonymy and polysemy.

In this section we explain our work on this behalf, towards a dataset of scientific documents for semantic similarity. We used the Wikipedia redirec-

tions matching extracted keyphrases from abstracts in the ArnetMiner [7].

8.5.1 Wikipedia redirections

Wikipedia redirections are page identifiers created to help navigate along the articles in Wikipedia. Basically, a page identifier is a *title*, with spaces replaced by underscores, to an article's page and a *redirect* just turns a reader to a target page. Redirects are created because readers may search for an article under different names or editors think that a given article should be named differently[8]. What is important for our work is the fact that redirects in Wikipedia are created by human editors, also, given that they are written as short titles it is likely that they match the name of important terms. Thus, it is a large dataset of semantic information with relations between titles or multi-word terms constantly curated by human editors.

The redirections can be retrieved from the Wikipedia data dumps [9]. The description for all the available formats and data dumps can be found in the official site for the Wikipedia Data Dumps[10].

According to the Wikipedia's documentation there are many reasons to create redirects:

- Alternative names for the same thing.
- Alternative spellings, capitalizations, etc.
- Common misspellings.
- Plurals.
- Subtopics that don't have their own article.
- Shortcuts to a page.
- Keeping links after it has been renamed.

These different scenarios to create a redirect from a page's title to another can be observed in Table 8.1. It shows a small list of the pages' ids without the underscores and their corresponding redirects. They are taken from the

[7] ArnetMiner Dataset https://aminer.org/data
[8] Wikipedia redirections https://en.wikipedia.org/wiki/Help:Redirect
[9] Wikimedia Downloads https://dumps.wikimedia.org/backup-index.html
[10] Wikipedia Data Dumps https://meta.wikimedia.org/wiki/Data_dumps/What%27s_available_for_download#Database_tables

database data dump 20180801 (August 1st of 2018) of the English Wikipedia, concretely, from the SQL tables *"redirect"*[11] and *"page"*[12].

Table 8.1 *Example of* Wikipedia redirections.

Titles (Pages)	*Redirects to:*
Empire Awards 2008 2008 Empire Awards	13th Empire Awards
European Film Awards 2000	13th European Film Awards
13th Field Artillery Regiment (United States) 13th Artillery Regiment (United States)	13th Field Artillery Regiment
13 FS 13th Fighter Squadron (United States) 13th Tactical Fighter Squadron 13th Expeditionary Fighter Squadron 13th TFS 13th Tactical Fighter Training Squadron	13th Fighter Squadron
13th Fighter Squadron (Disambiguation)	13th Fighter Squadron (disambiguation)
13th Flying Training Wing (JASDF)	13th Flight Training Wing (JASDF)
13th Flying Broom International Women's Film Festival	13th Flying Broom International Women's Film Festival
13th G8 summit 1987 G7 summit	13th G7 summit
1998 Gemini Awards 1998 Gemini Award	13th Gemini Awards

Take into consideration, that page's titles in Wikipedia are mostly from articles about general knowledge, not all of them are used in a technical context as we require. However, there are titles matching terms that we can use, for example, in Table 8.2 is shown a list of redirects to the page for the term *"Entropy"*. Looking for any of those titles in Wikipedia should redirect to the page for the term *"Entropy"*. Observe that some of the variations correspond to misspellings, e.g. the missing space in *"Disorder(thermodynamics)"*.

Although, the redirects in Wikipedia could help us to find lexical variations of the names of terms in scientific documents, we can not consider those terms a *synonyms* because of the other reasons to the creation of a redirect, in other words, the redirection of *subtopics* to a target page, the misspellings and the shortcuts. The redirections of subtopics can be hard to spot, consid-

[11]Description of the SQL table *"redirect"* https://www.mediawiki.org/wiki/Manual:Redirect_table

[12]Description of the SQL table *"page"* https://www.mediawiki.org/wiki/Manual:Page_table

Table 8.2 *Example of titles redirecting to the page for the term* "Entropy".

Target page: *Entropy*	
Entropic	Molar entropy
Entropy (thermodynamics)	Entropies
About Entropy	Entropical
Entropically favorable	Entropically
Thermodynamic entropy	Entropie
Entropy unit	Entrophy
Disorder (thermodynamics)	Specific entropy
Disorder(thermodynamics)	Acc3ss
Etropy	Antropy
Entropy change	Delta s
Enthropy	Kku/Books/Entropy
Entropy (general concept)	Entropy and Expansion of Universe

ering that we did not find useful field in the Wikipedia database to make the distinction, we think that it might be needed human intervention to label the true *synonyms*.

The *polysemy* is constantly present in the redirections, for example, there are additional redirections to those in Table 8.2. These are variations of *"Entropy"*, shown in Table 8.3, depending on the domain in which the term is being used. Observe that there are many redirections to the term *"Entropy"* in the fields of *"(statistical thermodynamics)"* and *"(information theory)"*. Both variations are referred with the single-word term *"Entropy"*, but distinguished with the attachment of the field inside parenthesis, hence the lexical representation is not the same. Of course, the additional text can be trunked but it will increase the number of false synonyms. The complete list of variations to the term *"Entropy"* is shown in the Table B.1 in Appendix B.

There are cases in which *"polysemy"* is not present, for example, the redirects shown in Table 8.4, there are not different meanings of the term *"Logistic regression"*, though there are *subtopics*, e.g. *"Logit model"* and *"Binary logit model"*.

Even though there are disadvantages, i.e. subtopics and polysemy, using the Wikipedia redirections allow us to select a subset of documents with semantic term relations even if they do not share the same lexical representation, easing the human annotation. It is a step towards the generation of a dataset as we require.

Table 8.3 *Other redirections to the term* "Entropy".

Page	*Target page*	*Page*	*Target page*
Thermodynamic entropy	Entropy (disambiguation)	Information entropy	Entropy (information theory) –Continuation–
Mathematical entropy		Entropy of a probability distribution	
Entropy (mathematics)		Entropy (statistics)	
Entropy (board game)		Weighted entropy	
Shannon entropy	Entropy (information theory)	Shannon Entropy	
Information Theoretic Entropy		Disordered state	Entropy (order and disorder)
Informational entropy		Boltzmann principle	Entropy (statistical thermodynamics)
Infotropy		Gibbs entropy	
Shannon information		Statistical entropy	
Shannon's entropy		Entropy (statistical views)	
Average information		Boltzmann-Gibbs entropy	
Information Entropy		Gibbs entropy formula	

Table 8.4 *Redirections for the term* "Logistic regression".

Page	*Target page*
Logit model	Logistic regression
Logit regression	
Binary logit model	
Logistic regression models	
Conditional logit analysis	

Matching redirects with keyphrases

We matched the Wikipedia redirections, described in this section, with the keyphrases within the text of 90% of the abstracts of the documents in the the Citation Network (Tang et al. 2008)[13] from the ACM V9 dataset in ArnetMiner. The rest of the abstracts (10% of the dataset) is left as testing data for future reference. The general information about the Wikipedia redirections and the extracted keyphrases is shown in Table 8.5.

We extracted keyphrases from the abstracts using Kleis[14]. In order to increase the recall of the extracted keyphrases the pre-trained model for the SemEval 2017 Task 10 using the parameters to include the PoS tag sequences as features and trained only with candidates filtered with at least 10 occurrences

[13]Citation Network in the AMiner Datasethttps://aminer.org/citation
[14]Kleis v0.1.2.dev0 https://github.com/sdhdez/kleis-keyphrase-extraction/releases/tag/v0.1.2.dev0

Table 8.5 *Information from Wikipedia redirections and extracted keyphrases.*

Wikipedia redirections (ACM V9)	Wikipedia pages' ids (titles)	11,523,073
	Redirections (target pages)	3,394,754
	Document's ids (citation network)	2,385,066
Extracted keyphrases	Documents processed with Kleis	2,146,777
	Documents with extracted keyphrases	1,475,448
	Total of keyphrases (normalized)	3,922,202
	Keyphrases per document processed	1.83 (mean)
	Keyphrases per document with keyphrases	2.66 (mean)

in the corpus, i.e. features_method="simple-posseq", filter_min_count=10. All the extracted keyphrases and redirections were normalized, lowercase without underscores ("_"), for example, the page's id *"Logit_regression"* is normalized as *"logit regression"*. Tables 8.6 and 8.7 show examples of the extracted keyphrases from the abstracts matching pages' ids from Wikipedia.

Table 8.6 *Most common keyphrases (or pages' ids) in the ACM V9.*

Keyphrase	*No. of documents*	*Keyphrase*	*No. of documents*
algorithm	1870	article	1163
algorithms	1525	editorial	1081
solution	1495	message	1019
simulation	1408	computers	1016
data	1396	computation	963
system	1354	g	933
information	1292	copyright page	856
problem	1226	addition	838
preface	1214	finite	794
users	1210	performance	778

We obtained the following information from the match of the extracted keyphrases and the redirections[15].

- 226,310 documents with keyphrases matching pages' ids.
- 218,637 keyphrases matching pages' ids.
- 125,735 concepts with more than one lexical representation (pages' ids) or related terms (subtopics) in the documents.
- 150,968 target pages (pages' ids matching keyphrases).

[15]URL to download the generated datasets https://github.com/sdhdez/matching-keyphrases-and-wikipedia-redirects.git

Table 8.7 *Keyphrases (or pages' ids) occurring in 2 documents of the ACM V9.*

Keyphrase	*Keyphrase*
clock frequencies	rhythmicity
avg	silicon-germanium-on-insulator
scalar field	licensure
efi	numerical value
oceania	refutably
energy norm	malhotra
reversing	mean value analysis
poisson random variable	sre
software development methodology	cost-benefit ratio
cerebral aneurysms	spatial decision support system

- 41,292 target pages with more than one lexical representation.
- 52,132 documents containing concepts with more than one lexical representation, or related terms.
- 39,985 documents containing only one form of the possible representations of a concept.

Chapter 9

Contributions and future work

In this part of our work, we analyzed the Pearson's correlation of different measures of document similarities (Jaccard similarity and cosine similarity), using their lexical content, i.e. bag of words and the extracted keyphrases, and word embeddings, e.g. centroids of word embeddings. We worked with a sample of abstracts from the scientific papers in the Citation Network (ACM V9) from ArnetMiner. It is the same dataset that we used later in our work towards the generation of a dataset of scientific documents to measure semantic similarity.

We achieved to select a subset of abstracts, from 52,132 scientific papers, with semi-automatic annotations of related terms and synonyms. We consider the annotations as *semi-automatic* because we extracted keyphrases from the abstracts using our system Kleis. These extracted keyphrases were matched with the Wikipedia redirections, which are human annotated relations between titles of Wikipedia pages.

Plus, from the 52,132 abstracts we selected a subset of 39,985 abstracts in which there is only one representation of a concept. Meaning that there are documents addressing the same concept, although each document uses a different lexical representation or at least they refer to a subtopic of the document. These annotations could be used to explore the effectiveness of semantic similarity measures on documents in the presence of related terms and synonyms.

9.1 Conclusions

We verified that documents with very common vocabulary are more correlated to all their pairs, both lexically and semantically, than those documents containing a less common vocabulary. This behavior is expected because the

centroids of word embeddings that we used as document representations depend on the document content, which is biasing the measurement. This means that those documents containing very common vocabulary are hard to characterize semantically. However, there are outliers in which there is almost no correlation between the content words and the semantic similarity, we identified that in some cases it is a consequence of the document not being written in English, even though, we think that this kind of outliers deserve more analysis.

We worked under the assumption, by definition, that keyphrases are the most relevant terms in a document, thus they should be providing more semantic information than single words. At the beginning of the experiments, we expected to find a better correlation between the similarity measures using the *bag of keyphrases* and the *centroids of word embeddings from the keyphrases*, after all, both are supposed to be sources of semantic information. However, it is not the case, in fact the use of both measurements present the worst correlation of all. We think that it might be a resulting bias of the method that we used to obtain the centroids of word embeddings, considering that it relies on the separated words conforming the keyphrases.

We confirmed that the measurement of similarity with keyphrases can be used for dimensionality reduction, to a certain extend, when a document is represented by centroids of word embeddings. Seeing that they correlate similarly to the the centroids from the *bag of words*. Though, we should not forget that the centroids are a very general representation of a document and might not be the best document representation, in view of the correlations showing that they are high dependable on the vocabulary. We also need to experiment with different keyphrase extractors to confirm that the behavior is not just related to the quality of our keyphrase extraction.

At the moment and as for our knowledge, there is not a dataset available to evaluate the semantic similarity on scientific publications in the presence of synonymy and polysemy. Having a dataset to evaluate the effectiveness of any semantic measure is essential to continue with our work. We achieved steps towards a dataset of scientific documents to evaluate semantic similarity, however it still needs human verification and more work to be ready to evaluate the semantic retrieval of documents to improve the search of the *state of the art*.

9.2 Discussion

As we wrote in several occasions, the annotation of a dataset of scientific documents is not easy because it needs high specialization on the topics ad-

dressed by the documents. What we can do, is to ease the annotation, for example, a possible option is to clusterize the documents in our dataset using the keywords from the metadata of the articles in the Citation Network. The resulting groups can be helpful to manually check the semi-automatic annotations that we included in the documents. At the same time, we could use the clusters to compare the effectiveness of using extracted keyphrases and different semantic similarity measures. However, it is difficult to define how many groups to consider and it would be needed to evaluate the clustering considering lexical and semantic features.

The objective of creating a dataset with our required attributes can be helpful to evaluate similarity measures useful to improve the retrieval of the *state of the art*. We also think that it should allow us to determine which measures, features or document representations are less dependent on the vocabulary and more reliable semantically.

9.3 Perspectives and future work

We are working to extend the comparison of the correlation of the vocabulary in the documents and the semantic information from the document representation or the features. Currently, we are experimenting with the Spearman's and Kendall's correlation coefficient and we are extending our experiments to test document vectors without the centroids of word embeddings, e.g. LDA, LSA, PPMI. For future work we want to experiment with our own model of word and keyphrase embeddings to evaluate and compare their performance.

In order to generate a dataset of scientific documents to measure semantic similarity we need to label the current annotations from the Wikipedia redirections, between related concepts (subtopics) and synonyms. We also are planning to extend the annotations to label term ambiguity with the help of redirections, however it is a more difficult task.

Publications

List of publications

SemEval 2017 Task 10

"LIPN at SemEval-2017 Task 10: Filtering Candidate Keyphrases from Scientific Publications with Part-of-Speech Tag Sequences to Train a Sequence Labeling Model" (Hernandez et al. 2017a), it is the system description of our participation on SemEval 2017 Task 10.

The team participated in the task under the name LIPN in Scenario 1, that includes three subtasks, Identification of keyphrases (Subtask A), Classification of identified keyphrases (Subtask B) and Extraction of relationships between two identified keyphrases (Subtask C).

Table 9.1 *Results for team LIPN in Scenario 1 at SemEval 2017 Task 10.*

Subtasks	Precision	Recall	$F-1$ score
A	0.31	0.49	0.38
A, B	0.17	0.27	0.21
C	0.33	0.02	0.05
A, B, C (Scenario 1)	0.17	0.25	0.21

Subtasks A and B were addressed as sequence labeling problems using Conditional Random Fields (CRFs). Subtask C was out of the scope of the approach, still we included one rule to relate possible synonyms without getting a significant result.

The presented system was mainly focused on the use of sequences of part-of-speech tags to filter candidate phrases for keyphrase identification, it is the approach described in this work but with some variations in the used features. [1]

[1] We used python-crfsuite with the default parameters for Named Entity Recognition, 'c1': 1.0, 'c2': 1e-3, 'max_iterations': 50, 'feature.possible_transitions': True, https://github.com/scrapinghub/python-crfsuite

EMC-Sci

"Part-of-Speech Tag Sequences to Filter Candidates for Keyphrase Extraction from Scientific Publications" (Hernandez et al. 2017b), it is the regular paper submitted to the *1st Workshop on Extracting and Modelling Scientific Knowledge from Texts* (EMC-Sci).

In this paper were described results on automatic keyphrase extraction, obtained from the preliminary experiments for the system presented at SemEval 2017 Task 10, those experiments were made using the approach described in this work, using *sequences of part-of-speech tags* (*PoS sequences*) to filter candidate phrases to train a Conditional Random Field model.

This paper presents experiments for keyphrase identification using *sequences of part-of-speech tags* (*PoS sequences*) to filter candidate keyphrases used to train a Conditional Random Fields model. These experiments are the base of the system presented by the team LIPN at *SemEval 2017 Task 10: Extracting Keyphrases and Relations from Scientific Publications*.

VADOR 2017

"Classification of Keyphrases from Scientific Publications Using WordNet and Word Embeddings" (Buscaldi et al. 2017), it is the paper submitted to the *1er atelier Valorisation et Analyse des Données de la Recherche* (VADOR 2017).

The ScienceIE task at SemEval-2017 introduced an epistemological classification of keyphrases in scientific publications, suggesting that research activities revolve around the key concepts of *process* (methods and systems), *material* (data and physical resources) and *task*.

In this paper we present a method for the classification of keyphrases according to the ScienceIE classification, using WordNet and word embeddings derived features. The method outperforms the best system at SemEval-2017, although our experiments highlighted some issues with the collection.

The method we propose in this paper is based on Support Vector Machines (SVM), in particular the nu-SVM implementation by (Chang and Lin 2011). SVMs are well known maximum margin classifiers; we chose them because of their robustness with regard to problems with a large number of features. Please note that the method we are describing in this paper only shares part of the WordNet-based features with the one we used to participate to the task (Hernandez et al. 2017a).

Appendix A

Part-of-speech tags from PerceptronTagger in NLTK

Part-of-speech tags from the CoNLL data, (WSJ part of Penn Treebank,) present in the pre-trained model of the PerceptronTagger[1] in NLTK 3.4. The descriptions and examples in Table A.1 are found in the help module from NLTK *"Help on tag sets"*[2]. The list was obtained using the following Python code.

```
import nltk
nltk.help.upenn_tagset()
```

Table A.1 *List of part-of-speech tags used by the model in PerceptronTagger in NLTK 3.4.*

Tag	*Description*	*Example*
#	#	
$	$	
”	closing quotation mark	’ ”
(	opening parenthesis	([{
)	closing parenthesis	)] }
,	comma	,
.	sentence terminator	. ! ?
:	colon or ellipsis	: ; ...
CC	conjunction, coordinating	& ’n and both but either et for less minus neither nor or plus so therefore times v. versus vs. whether yet

– *Continued on next page*

[1] PerceptronTagger module from NLTK 3.4 http://www.nltk.org/api/nltk.tag.html#nltk.tag.perceptron.PerceptronTagger

[2] List of NLTK Corpora http://www.nltk.org/nltk_data/

Table A.1 – continued from previous page

Tag	*Description*	*Example*
CD	numeral, cardinal	mid-1890 nine-thirty forty-two one-tenth ten million 0.5 one forty-seven 1987 twenty '79 zero two 78-degrees eighty-four IX '60s .025 fifteen 271,124 dozen quintillion DM2,000 ...
DT	determiner	all an another any both del each either every half la many much nary neither no some such that the them these this those
EX	existential there	there
FW	foreign word	gemeinschaft hund ich jeux habeas Haementeria Herr K'ang-si vous lutihaw alai je jour objets salutaris fille quibusdam pas trop Monte terram fiche oui corporis ...
IN	preposition or conjunction, subordinating	astride among uppon whether out inside pro despite on by throughout below within for towards near behind atop around if like until below next into if beside ...
JJ	adjective or numeral, ordinal	third ill-mannered pre-war regrettable oiled calamitous first separable ectoplasmic battery-powered participatory fourth still-to-be-named multilingual multi-disciplinary ...
JJR	adjective, comparative	bleaker braver breezier briefer brighter brisker broader bumper busier calmer cheaper choosier cleaner clearer closer colder commoner costlier cozier creamier crunchier cuter ...
JJS	adjective, superlative	calmest cheapest choicest classiest cleanest clearest closest commonest corniest costliest crassest creepiest crudest cutest darkest deadliest dearest deepest densest dinkiest ...
LS	list item marker	A A. B B. C C. D E F First G H I J K One SP-44001 SP-44002 SP-44005 SP-44007 Second Third Three Two * a b c d first five four one six three two
MD	modal auxiliary	can cannot could couldn't dare may might must need ought shall should shouldn't will would
NN	noun, common, singular or mass	common-carrier cabbage knuckle-duster Casino afghan shed thermostat investment slide humour falloff slick wind hyena override subhumanity machinist ...
NNP	noun, proper, singular	Motown Venneboerger Czestochwa Ranzer Conchita Trumplane Christos Oceanside Escobar Kreisler Sawyer Cougar Yvette Ervin ODI Darryl CTCA Shannon A.K.C. Meltex Liverpool ...
NNPS	noun, proper, plural	Americans Americas Amharas Amityvilles Amusements Anarcho-Syndicalists Andalusians Andes Andruses Angels Animals Anthony Antilles Antiques Apache Apaches Apocrypha ...
NNS	noun, common, plural	undergraduates scotches bric-a-brac products bodyguards facets coasts divestitures storehouses designs clubs fragrances averages subjectivists apprehensions muses factory-jobs ...
PDT	pre-determiner	all both half many quite such sure this

– Continued on next page

Table A.1 – continued from previous page

Tag	*Description*	*Example*
POS	genitive marker	' 's
PRP	pronoun, personal	hers herself him himself hisself it itself me myself one oneself ours ourselves ownself self she thee theirs them themselves they thou thy us
PRP$	PRP$	
RB	adverb	occasionally unabatingly maddeningly adventurously professedly stirringly prominently technologically magisterially predominately swiftly fiscally pitilessly ...
RBR	adverb, comparative	further gloomier grander graver greater grimmer harder harsher healthier heavier higher however larger later leaner lengthier less-perfectly lesser lonelier longer louder lower more ...
RBS	adverb, superlative	best biggest bluntest earliest farthest first furthest hardest heartiest highest largest least less most nearest second tightest worst
RP	particle	aboard about across along apart around aside at away back before behind by crop down ever fast for forth from go high i.e. in into just later low more off on open out over per pie raising start teeth that through under unto up up-pp upon whole with you
SYM	symbol	% & ' " ".)). * + ,. < = > @ A[fj] U.S U.S.S.R * ** ***
TO	"to" as preposition or infinitive marker	to
UH	interjection	Goodbye Goody Gosh Wow Jeepers Jee-sus Hubba Hey Kee-reist Oops amen huh howdy uh dammit whammo shucks heck anyways whodunnit honey golly man baby diddle hush sonuvabitch ...
VB	verb, base form	ask assemble assess assign assume atone attention avoid bake balkanize bank begin behold believe bend benefit bevel beware bless boil bomb boost brace break bring broil brush build ...
VBD	verb, past tense	dipped pleaded swiped regummed soaked tidied convened halted registered cushioned exacted snubbed strode aimed adopted belied figgered speculated wore appreciated contemplated ...
VBG	verb, present participle or gerund	telegraphing stirring focusing angering judging stalling lactating hankerin' alleging veering capping approaching traveling besieging encrypting interrupting erasing wincing ...
VBN	verb, past participle	multihulled dilapidated aerosolized chaired languished panelized used experimented flourished imitated reunifed factored condensed sheared unsettled primed dubbed desired ...
VBP	verb, present tense, not 3rd person singular	predominate wrap resort sue twist spill cure lengthen brush terminate appear tend stray glisten obtain comprise detest tease attract emphasize mold postpone sever return wag ...

– Continued on next page

Table A.1 – continued from previous page

Tag	*Description*	*Example*
VBZ	verb, present tense, 3rd person singular	bases reconstructs marks mixes displeases seals carps weaves snatches slumps stretches authorizes smolders pictures emerges stockpiles seduces fizzes uses bolsters slaps speaks pleads ...
WDT	WH-determiner	that what whatever which whichever
WP	WH-pronoun	that what whatever whatsoever which who whom whosoever
WP$	WP$	
WRB	Wh-adverb	how however whence whenever where whereby whereever wherein whereof why
“	opening quotation mark	‘ “

Appendix B

List of all the *"entropy"* redirections

List of Wikipedia redirections from the SQL tables *"redirect"*[1] and *"page"*[2] in the database data dump 20180801 (August 1st of 2018) of the English Wikipedia.

Table B.1 *Full list of Wikipedia redirections containing the word* "entropy".

Pages (Titles)	*Redirect to:*
Votes for deletion/Entropy (linguistics)	Articles for deletion/Entropy (linguistics)
Votes for deletion/Entropy five	Articles for deletion/Entropy five
Cross-Entropy Method	Cross-Entropy Method
Cross entropy method	Cross-entropy method
Cross-Entropy Method	Cross-entropy method
Cross-entropy	Cross entropy
Mollier diagram	Enthalpy–entropy chart
H-s chart	Enthalpy–entropy chart
Mollier-diagram	Enthalpy–entropy chart
Enthalpy-entropy chart	Enthalpy–entropy chart
H–s chart	Enthalpy–entropy chart
Entropy-enthalpy compensation	Enthalpy–entropy compensation
Enthalpy-entropy compensation	Enthalpy–entropy compensation
Entropic	Entropy
Entropy (thermodynamics)	Entropy

– *Continued on next page*

[1]Description of the SQL table *"redirect"* https://www.mediawiki.org/wiki/Manual:Redirect_table
[2]Description of the SQL table *"page"* https://www.mediawiki.org/wiki/Manual:Page_table

Table B.1 – continued from previous page

Pages (Titles)	*Redirect to:*
About Entropy	Entropy
Entropically favorable	Entropy
Thermodynamic entropy	Entropy
Entropy unit	Entropy
Disorder (thermodynamics)	Entropy
Disorder(thermodynamics)	Entropy
Etropy	Entropy
Entropy change	Entropy
Enthropy	Entropy
Entropy (general concept)	Entropy
Molar entropy	Entropy
Entropies	Entropy
Entropical	Entropy
Entropically	Entropy
Entropie	Entropy
Entrophy	Entropy
Specific entropy	Entropy
Acc3ss	Entropy
Antropy	Entropy
Delta s	Entropy
Kku/Books/Entropy	Entropy
Entropy and Expansion of Universe	Entropy
Getrandom	Entropy-supplying system calls
Getentropy	Entropy-supplying system calls
Acc3ss/Archive1	Entropy/Archive1
Acc3ss/Archive 1	Entropy/Archive 1
Entropy/Archive1	Entropy/Archive 1
Entropy/Archive10	Entropy/Archive 10
Entropy/Archive11	Entropy/Archive 11
Entropy/Archive12	Entropy/Archive 12
Entropy/Archive13	Entropy/Archive 13
Entropy/Archive2	Entropy/Archive 2
Entropy/Archive3	Entropy/Archive 3
Entropy/Archive4	Entropy/Archive 4
Entropy/Archive5	Entropy/Archive 5
Entropy/Archive6	Entropy/Archive 6

– *Continued on next page*

Table B.1 – continued from previous page

Pages (Titles)	*Redirect to:*
Entropy/Archive7	Entropy/Archive 7
Entropy/Archive8	Entropy/Archive 8
Entropy/Archive9	Entropy/Archive 9
Acc3ss/Header	Entropy/Header
Acc3ss/huggle.css	Entropy/huggle.css
Acc3ss/monobook.css	Entropy/monobook.css
Entropy: Into a Greenhouse World	Entropy: A New World View
Hyle (board game)	Entropy (1977 board game)
Entropy (Buffy)	Entropy (Buffy the Vampire Slayer)
Entropy (Buffy episode)	Entropy (Buffy the Vampire Slayer)
Entropy (Buffy the Vampire Slayer episode)	Entropy (Buffy the Vampire Slayer)
Entropy (albums)	Entropy (EP)
Entropy (album)	Entropy (EP)
Entropy (Hip Hop Reconstruction from the Ground up)	Entropy (Hip Hop Reconstruction from the Ground Up)
Entropy (network)	Entropy (anonymous data store)
Thermodynamic arrow of time	Entropy (arrow of time)
The thermodynamic arrow of time	Entropy (arrow of time)
Cosmological entropy	Entropy (arrow of time)
Thermodynamic arrow	Entropy (arrow of time)
Entropy (thermodynamic views)	Entropy (classical thermodynamics)
Entropy (Linux)	Entropy (computing)
Entropy (GNU/Linux)	Entropy (computing)
Entropy (Unix)	Entropy (computing)
Thermodynamic entropy	Entropy (disambiguation)
Mathematical entropy	Entropy (disambiguation)
Entropy (mathematics)	Entropy (disambiguation)
Entropy (board game)	Entropy (disambiguation)
Energy dispersal	Entropy (energy dispersal)
Entropy (movie)	Entropy (film)
Shannon entropy	Entropy (information theory)
Information Theoretic Entropy	Entropy (information theory)
Informational entropy	Entropy (information theory)
Infotropy	Entropy (information theory)
Shannon information	Entropy (information theory)
Shannon's entropy	Entropy (information theory)
Average information	Entropy (information theory)

– Continued on next page

Table B.1 – continued from previous page

Pages (Titles)	*Redirect to:*
Information Entropy	Entropy (information theory)
Entropy (information theory)/Comments	Entropy (information theory)
Entropy (information)	Entropy (information theory)
Information entropy	Entropy (information theory)
Entropy (Information theory)	Entropy (information theory)
Entropy of a probability distribution	Entropy (information theory)
Entropy (statistics)	Entropy (information theory)
Entropy (information theory	Entropy (information theory)
Data compression/entropy	Entropy (information theory)
Data entropy	Entropy (information theory)
Weighted entropy	Entropy (information theory)
Shannon Entropy	Entropy (information theory)
Disordered state	Entropy (order and disorder)
Boltzmann principle	Entropy (statistical thermodynamics)
Gibbs entropy	Entropy (statistical thermodynamics)
Statistical entropy	Entropy (statistical thermodynamics)
Entropy (statistical views)	Entropy (statistical thermodynamics)
Boltzmann-Gibbs entropy	Entropy (statistical thermodynamics)
Gibbs entropy formula	Entropy (statistical thermodynamics)
Entropy (Hip Hop Reconstruction from the Ground Up)	Entropy / Send Them
Entropy (Hip Hop Reconstruction from the Ground up)	Entropy / Send Them
Capacity for entropy	Entropy and life
Entropy coder	Entropy encoding
Entropy coding	Entropy encoding
Entropy code	Entropy encoding
Minimum redundancy coding	Entropy encoding
Entropy coded	Entropy encoding
Entropy Estimation	Entropy estimation
Szilard engine	Entropy in thermodynamics and information theory
Zeilinger's principle	Entropy in thermodynamics and information theory
Zeilinger's principle	Entropy in thermodynamics and information theory
Szilard's engine	Entropy in thermodynamics and information theory

– Continued on next page

Table B.1 – continued from previous page

Pages (Titles)	*Redirect to:*
Entropy (anesthesiology)	Entropy monitoring
Entropy (monitor)	Entropy monitoring
Entropy Network	Entropy network
Activation entropy	Entropy of activation
Standard entropy change of fusion	Entropy of fusion
Gibbs theorem	Entropy of mixing
Gibbs free energy of mixing	Entropy of mixing
Identifying molecules in given locations	Entropy of mixing
Standard entropy change of vaporization	Entropy of vaporization
Source information rate	Entropy rate
Eternal-Entropy/Misc/Header	Eternal-Entropy/Header
Eternal-Entropy/intro	Eternal-Entropy/about
4-entropy	Four-vector
Max-entropy	Hartley function
MEMM	Maximum-entropy Markov model
Conditional Markov model	Maximum-entropy Markov model
Maximum entropy Markov model	Maximum-entropy Markov model
Min entropy	Min-entropy
Anirudh215/sandbox/Min entropy	Min-entropy
Minimal entropy martingale measure	Minimal-entropy martingale measure
Machintas	Ms.Entropy
Neg-Entropy	Negentropy
Self-entropy	Self-information
Temperature-entropy diagram	Temperature vs. specific entropy diagram
Temperature–entropy diagram	Temperature vs. specific entropy diagram
Oh7oh7oh7/Books/Entropy	Waterbug89/Books/Entropy

Bibliography

Agirre, E., Banea, C., Cardie, C., Cer, D., Diab, M., González-Agirre, A., Guo, W., Lopez-Gazpio, I., Maritxalar, M., Mihalcea, R., Rigau, G., Uria, L., & Wiebe, J. (2015). SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015). DOI: 10.18653/v1/S15-2045

Agirre, E., Banea, C., Cardie, C., Cer, D., Diab, M., González-Agirre, A., Guo, W., Mihalcea, R., Rigau, G., & Wiebe, J. (2014). SemEval-2014 Task 10: Multilingual Semantic Textual Similarity. Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). DOI: 10.3115/v1/S14-2010

Agirre, E., Banea, C., Cer, D., Diab, M., González-Agirre, A., Mihalcea, R., Rigau, G., & Wiebe, J. (2016). SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation. Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). DOI: 10.18653/v1/S16-1081

Agirre, E., Cer, D., Diab, M., & González-Agirre, A. (2012). SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. *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). Retrieved June 12, 2019, from https://www.aclweb.org/anthology/papers/S/S12/S12-1051/

Agirre, E., Cer, D., Diab, M., González-Agirre, A., & Guo, W. (2013). *SEM 2013 shared task: Semantic Textual Similarity. Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity. Retrieved June 12, 2019, from https://www.aclweb.org/anthology/papers/S/S13/S13-1004/

Alexandrov, M., Gelbukh, A., & Rosso, P. (2005). An Approach to Clustering Abstracts (A. Montoyo, R. Muńoz, & E. Métais, Eds.). In A. Montoyo, R. Muńoz, & E. Métais (Eds.), *Natural Language Processing and Information Systems*, Berlin, Heidelberg, Springer. DOI: 10.1007/11428817_25

Augenstein, I., Das, M., Riedel, S., Vikraman, L., & McCallum, A. (2017). SemEval 2017 Task 10: ScienceIE - Extracting Keyphrases and Relations from Scientific Publications, 546–555. http://aclweb.org/anthology/S17-2091

Baroni, M., Dinu, G., & Kruszewski, G. (2014). 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)*. DOI: 10.3115/v1/P14-1023

Baroni, M., & Lenci, A. (2010). Distributional Memory: A General Framework for Corpus-Based Semantics. *Computational Linguistics*, *36*(4), 673–721. DOI: 10.1162/coli_a_00016

Bengio, Y., Schwenk, H., Senécal, J.-S., Morin, F., & Gauvain, J.-L. (2006). Neural Probabilistic Language Models (D. E. Holmes & L. C. Jain, Eds.). In D. E. Holmes & L. C. Jain (Eds.), *Innovations in Machine Learning: Theory and Applications*. Berlin, Heidelberg, Springer Berlin Heidelberg. DOI: 10.1007/3-540-33486-6_6

Berend, G. (2011). Opinion Expression Mining by Exploiting Keyphrase Extraction, In *Proceedings of 5th International Joint Conference on Natural Language Processing*, Chiang Mai, Thailand, Asian Federation of Natural Language Processing. Retrieved March 12, 2019, from http://aclweb.org/anthology/I11-1130

Berry, M. W., Dumais, S. T., & O'Brien, G. W. (1995). Using Linear Algebra for Intelligent Information Retrieval. *SIAM Rev.*, *37*(4), 573–595. DOI: 10.1137/1037127

Bhaskar, P., Nongmeikapam, K., & Bandyopadhyay, S. (2012). Keyphrase Extraction in Scientific Articles: A Supervised Approach, In *Proceedings of COLING 2012: Demonstration Papers*, Mumbai, India, The COLING 2012 Organizing Committee. Retrieved January 17, 2019, from http://aclweb.org/anthology/C12-3003

Bjork, B.-C., Roos, A., & Lauri, M. (2009). Scientific Journal Publishing: Yearly Volume and Open Access Availability. *Information Research: An International Electronic Journal*, *14*(1). Retrieved February 17, 2019, from https://eric.ed.gov/?id=EJ837278

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, *3*, 993–1022. Retrieved March 14, 2019, from http://dl.acm.org/citation.cfm?id=944919.944937

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146.

Bolelli, L., Ertekin, S., & Giles, C. (2006). Clustering Scientific Literature Using Sparse Citation Graph Analysis. *Knowledge Discovery in Databases: PKDD 2006*, *4213*, 30–41. DOI: 10.1007/11871637_8

Bollegala, D., Matsuo, Y., & Ishizuka, M. (2007). Measuring semantic similarity between words using web search engines, In *In Proceedings of WWW*.

Brochu, E., & de Freitas, N. (2003). "Name That Song!" A Probabilistic Approach to Querying on Music and Text (S. Becker, S. Thrun, & K. Obermayer, Eds.). In S. Becker, S. Thrun, & K. Obermayer (Eds.), *Advances in Neural Information Processing Systems 15*. MIT Press. Retrieved June 13, 2019, from http://papers.nips.cc/paper/2262-name-that-song-a-probabilistic-approach-to-querying-on-music-and-text.pdf

Brokos, G.-I., Malakasiotis, P., & Androutsopoulos, I. (2016). Using Centroids of Word Embeddings and Word Mover's Distance for Biomedical Document Retrieval in Question Answering, 114–118. http://nlp.cs.aueb.gr

Buscaldi, D., Hernandez, S. D., & Charnois, T. (2017). Classification of Keyphrases from Scientific Publications using WordNet and Word Embeddings, In *VADOR (Valorisation et Analyse des Données de la Recherche) 2017*, Toulouse, France. https://hal.archives-ouvertes.fr/hal-01659677

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., & Specia, L. (2017). SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). DOI: 10.18653/v1/S17-2001

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*(3), 27:1–27:27.

Cover, T., & Thomas, J. (2006). Elements of Information Theory 2nd edn (Hoboken, NJ, John Wiley & Sons).

Cramer, J. S. (2002). The origins of logistic regression.

Dagan, I., Lee, L., & Pereira, F. C. N. (1999). Similarity-Based Models of Word Cooccurrence Probabilities. *Machine Learning*, *34*(1), 43–69. DOI: 10.1023/A:1007537716579

De Boom, C., Van Canneyt, S., Demeester, T., & Dhoedt, B. (2016). Representation learning for very short texts using weighted word embedding aggregation. *Pattern Recognition Letters*, *80*, 150–156. DOI: 10.1016/j.patrec.2016.06.012

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, *41*(6), 391–407.

Dumais, S., Platt, J., Heckerman, D., & Sahami, M. (1998). Inductive Learning Algorithms and Representations for Text Categorization, In *Proceedings of the Seventh International Conference on Information and Knowledge Management*, Bethesda, Maryland, USA, ACM. DOI: 10.1145/288627.288651

Fano, R. M. (1961). Transmission of Information: A Statistical Theory of Communications. *American Journal of Physics*, *29*(11), 793–794. DOI: 10.1119/1.1937609

Firth, J. R. (1957). A synopsis of linguistic theory 1930-55., *1952-59*, 1–32.

Frank, E., Paynter, G. W., Witten, I. H., Gutwin, C., & Nevill-Manning, C. G. (1999). Domain-Specific Keyphrase Extraction, In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. http://dl.acm.org/citation.cfm?id=646307.687591

Gabrilovich, E., & Markovitch, S. (2007). Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis, In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, Hyderabad, India, Morgan Kaufmann Publishers Inc. Retrieved June 13, 2019, from http://dl.acm.org/citation.cfm?id=1625275.1625535

Ganesan, P., Garcia-Molina, H., & Widom, J. (2003). Exploiting Hierarchical Domain Structure to Compute Similarity. *ACM Trans. Inf. Syst.*, *21*(1), 64–93. DOI: 10.1145/635484.635487

Gollapalli, S. D., & Caragea, C. (2014). Extracting Keyphrases from Research Papers Using Citation Networks, In *Twenty-Eighth AAAI Conference on Artificial Intelligence*. Twenty-Eighth AAAI Conference on Artificial Intelligence. Retrieved March 12, 2019, from https://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8662

Gong, Z., & Liu, Q. (2009). Improving keyword based web image search with visual feature distribution and term expansion. *Knowledge and Information Systems*, *21*(1), 113–132. DOI: 10.1007/s10115-008-0183-x

Greene, D., & Cunningham, P. (2006). Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering, In *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, Pennsylvania, USA, ACM. DOI: 10.1145/1143844.1143892

Grineva, M., Grinev, M., & Lizorkin, D. (2009). Extracting key terms from noisy and multitheme documents, In *Proceedings of the 18th international conference on World wide web - WWW '09*, ACM Press. DOI: 10.1145/1526709.1526798

Gutwin, C., Paynter, G., Witten, I., Nevill-Manning, C., & Frank, E. (1999). Improving Browsing in Digital Libraries with Keyphrase Indexes. *Decis. Support Syst.*, *27*(1-2), 81–104. DOI: 10.1016/S0167-9236(99)00038-X

Haddoud, M., Mokhtari, A., Lecroq, T., & Abdeddaïm, S. (2015). Accurate Keyphrase Extraction from Scientific Papers by Mining Linguistic Information. *Proceedings of the First Workshop on Mining Scientific Papers: Computational Linguistics and Bibliometrics co-located with 15th International Society of Scientometrics and Informetrics Conference (ISSI 2015) Istanbul, Turkey, June 29, 2015.*, *1384*, 12–17. http://ceur-ws.org/Vol-1384/paper2.pdf

Hammouda, K. M., Matute, D. N., & Kamel, M. S. (2005). CorePhrase: Keyphrase extraction for document clustering. Proceedings of the 4th International Conference on Machine Learning and Data Mining in Pattern Recognition, Springer-Verlag. DOI: 10.1007/11510888_26

Harispe, S., Ranwez, S., Janaqi, S., & Montmain, J. (2015). Semantic Similarity from Natural Language and Ontology Analysis. *Synthesis Lectures on Human Language Technologies*, *8*(1), 1–254. DOI: 10.2200/S00639ED1V01Y201504HLT027

Harris, Z. (1954). Distributional structure. Word, 10 (2-3): 146–162. Reprinted in Fodor, J. A and Katz, JJ (eds.), Readings in the Philosophy of Language.

Hasan, K. S., & Ng, V. (2010). Conundrums in unsupervised keyphrase extraction: Making sense of the State-of-the-Art, In *Coling 2010 - 23rd International Conference on Computational Linguistics, Proceedings of the Conference*, Association for Computational Linguistics. https://dl.acm.org/citation.cfm?id=1944608

Hasan, K. S., & Ng, V. (2014). Automatic Keyphrase Extraction: A Survey of the State of the Art. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 12. DOI: 10.3115/v1/P14-1119

Hernandez, S. D., Buscaldi, D., & Charnois, T. (2017a). LIPN at SemEval-2017 Task 10: Filtering Candidate Keyphrases from Scientific Publications with Part-of-Speech Tag Sequences to Train a Sequence Labeling Model. *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 994–998. DOI: 10.18653/v1/S17-2174

Hernandez, S. D., Buscaldi, D., & Charnois, T. (2017b). Part-of-Speech Tag Sequences to Filter Candidates for Keyphrase Extraction from Scientific Publications, In *1er atelier sur l' Extraction et la Modélisation de Connaissances à partir de textes scientifiques (EMC-Sci)*. IC 2017, Caen, France. https://sites.google.com/view/emcsci/fran%C3%A7ais/programme

Hofmann, T. (1999). Probabilistic Latent Semantic Indexing, In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, California, USA, ACM. DOI: 10.1145/312624.312649

Holat, P., Tomeh, N., Charnois, T., Battistelli, D., Jaulent, M.-C., & Métivier, J.-P. (2016). Weakly-Supervised Symptom Recognition for Rare Diseases in Biomedical Text (H. Boström, A. Knobbe, C. Soares, & P. Papapetrou, Eds.). In H. Boström, A. Knobbe, C. Soares, & P. Papapetrou (Eds.), *Advances in Intelligent Data Analysis XV*, Springer International Publishing.

Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge, In *Proceedings of the 2003 conference on Empirical methods in natural language processing* -, Association for Computational Linguistics. DOI: 10.3115/1119355.1119383

Hulth, A., & Megyesi, B. B. (2006). A study on automatically extracted keywords in text categorization. *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL ACL 06*, 537–544. DOI: 10.3115/1220175.1220243

Kanerva, P., Kristoferson, J., & Holst, A. (2000). Random Indexing of Text Samples for Latent Semantic Analysis, In *In Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, Erlbaum.

Kim, S. N., & Kan, M.-Y. (2009). Re-examining automatic keyphrase extraction approaches in scientific articles. *Proceedings of the Workshop on Multiword Expressions Identification, Interpretation, Disambiguation and Applications - MWE '09*, 9–9. DOI: 10.3115/1698239.1698242

Kim, S. N., Medelyan, O., Kan, M.-Y., & Baldwin, T. (2010). SemEval-2010 Task 5: Automatic Keyphrase Extraction from Scientific Articles, In *Proceedings of the 5th International Workshop on Semantic Evaluation*, Los Angeles, California, Association for Computational Linguistics. Retrieved February 17, 2019, from http://dl.acm.org/citation.cfm?id=1859664.1859668

Kim, S. N., Medelyan, O., Kan, M.-Y., & Baldwin, T. (2013). Automatic keyphrase extraction from scientific articles. *Language Resources and Evaluation*, *47*(3), 723–742. DOI: 10.1007/s10579-012-9210-3

Kolb, P. (2009). Experiments on the difference between semantic similarity and relatedness, In *Proceedings of the 17th Nordic Conference of Computational Linguistics (NODALIDA 2009)*, Odense, Denmark, Northern European Association for Language Technology (NEALT). Retrieved November 7, 2019, from https://www.aclweb.org/anthology/W09-4613

Kusner, M. J., Sun, Y., Kolkin, N. I., & Weinberger, K. Q. (2015). From Word Embeddings To Document Distances. *Proceedings of The 32nd International Conference on Machine Learning*, *37*, 957–966. http://proceedings.mlr.press/v37/kusnerb15.html

Landauer, T. K. [Thomas K], & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, *104*(2), 211.

Landauer, T. K. [Thomas K.], Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, *25*(2-3), 259–284. DOI: 10.1080/01638539809545028

Larsen, P. O., & von Ins, M. (2010). The rate of growth in scientific publication and the decline in coverage provided by Science Citation Index. *Scientometrics*, *84*(3), pmid 20700371, 575–603. DOI: 10.1007/s11192-010-0202-z

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents, In *International conference on machine learning*.

Lenci, A. (2008). Distributional semantics in linguistic and cognitive research. *Italian Journal of Linguistics*, *20*.

Li, H., & Yamanishi, K. (2000). Topic Analysis Using a Finite Mixture Model, In *2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. http://aclweb.org/anthology/W00-1305

Lintean, M., Moldovan, C., Rus, V., & McNamara, D. (2010). The Role of Local and Global Weighting in Assessing the Semantic Similarity of Texts Using Latent Semantic Analysis, In *Twenty-Third International FLAIRS Conference*. Twenty-Third International FLAIRS Conference. Retrieved June 13, 2019, from https://www.aaai.org/ocs/index.php/FLAIRS/2010/paper/view/1310

Liu, F., Pennell, D., & Liu, Y. (2009). Unsupervised approaches for automatic keyword extraction using meeting transcripts. *Naacl-Hlt*, 620–628. DOI: 10.3115/1620754.1620845

Lund, K., & Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, *28*(2), 203–208. DOI: 10.3758/BF03204766

Manning, C., Raghavan, P., & Schütze, H. (2010). Introduction to Information Retrieval. *Natural Language Engineering*, *16*, 100–103. Retrieved February 18, 2019, from http://eprints.bimcoordinator.co.uk/35/

McCallum, A., & Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 -*, *4*, 188–191. DOI: 10.3115/1119176.1119206

McCallum, A., & Nigam, K. (1999). Text Classification by Bootstrapping with Keywords, EM and Shrinkage, In *Unsupervised Learning in Natural Language Processing*. http://aclweb.org/anthology/W99-0908

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into texts. *Proceedings of EMNLP*, *85*, 404–411. DOI: 10.3115/1219044.1219064

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, 3111–3119. https://dl.acm.org/citation.cfm?id=2999959#.W1iNlPjiiyQ.mendeley

Mikolov, T., Yih, W.-t., & Zweig, G. (2013). Linguistic Regularities in Continuous Space Word Representations. *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 746–751. https://aclanthology.info/papers/N13-1090/n13-1090

Miller, G. A. [George A]. (1995). WordNet: A lexical database for English. *Communications of the ACM*, *38*(11), 39–41.

Miller, G. A. [George A.], & Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes*, *6*(1), 1–28. DOI: 10.1080/01690969108406936

Mohammad, S. M., & Hirst, G. (2012). Distributional Measures as Proxies for Semantic Relatedness, arxiv 1203.1889. Retrieved November 5, 2019, from http://arxiv.org/abs/1203.1889

Naik, M. P., Prajapati, H. B., & Dabhi, V. K. (2015). A survey on semantic document clustering, In *Proceedings of 2015 IEEE International Conference on Electrical, Computer and Communication Technologies, ICECCT 2015*. DOI: 10.1109/ICECCT.2015.7226036

Nanba, H., Kando, N., & Okumura, M. (2011). Classification of research papers using citation links and citation types: Towards automatic review article generation. *Advances in Classification Research Online*, *11*(1), 117–134. DOI: 10.7152/acro.v11i1.12774

Nguyen, T. D., & Kan, M.-Y. (2007). Keyphrase Extraction in Scientific Publications (D. H.-L. Goh, T. H. Cao, I. T. Sølvberg, & E. Rasmussen, Eds.). In D. H.-L. Goh, T. H. Cao, I. T. Sølvberg, & E. Rasmussen (Eds.), *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, Springer Berlin Heidelberg.

Ontrup, J., & Ritter, H. (2001). Hyperbolic Self-Organizing Maps for Semantic Navigation, In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, Vancouver, British Columbia, Canada,

MIT Press. Retrieved June 13, 2019, from http://dl.acm.org/citation.cfm?id=2980539.2980723

Osborne, F., & Motta, E. (2015). Klink-2: Integrating Multiple Web Sources to Generate Semantic Topic Networks, In *Proceedings of the 14th International Conference on The Semantic Web - ISWC 2015 - Volume 9366*, New York, NY, USA, Springer-Verlag New York, Inc. DOI: 10.1007/978-3-319-25007-6_24

Pal, T., Banka, H., Mitra, P., & Das, B. (2011). Linguistic Knowledge Based Supervised Key-phrase Extraction, In *Proceedings of national conference on future trends in information & communication technology & applications, Bhubaneswar, India*.

Pedersen, T., Pakhomov, S. V. S., Patwardhan, S., & Chute, C. G. (2007). Measures of Semantic Similarity and Relatedness in the Biomedical Domain. *J. of Biomedical Informatics*, *40*(3), 288–299. DOI: 10.1016/j.jbi.2006.06.004

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation, In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations, In *Proc. of NAACL*.

Pinto, D., Jiménez-Salazar, H., & Rosso, P. (2006). Clustering abstracts of scientific texts using the transition point technique. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *3878 LNCS*, 536–546. DOI: 10.1007/11671299_55

QasemiZadeh, B., & Schumann, A.-K. (2016). The ACL RD-TEC 2.0: A Language Resource for Evaluating Term Extraction and Entity Recognition Methods. *Proceedings of the 10th edition of the Language Resources and Evaluation Conference*, 1862–1868. http://pars.ie/publications/papers/pre-prints/lrec2016_acl_rd_tec2.pdf

Quadrianto, N., Song, L., & Smola, A. J. (2009). Kernelized Sorting (D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou, Eds.). In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *Advances in Neural Information Processing Systems 21*. Curran Associates, Inc. Retrieved June 13, 2019, from http://papers.nips.cc/paper/3608-kernelized-sorting.pdf

Rangel, F., Rosso, P., Potthast, M., & Stein, B. (2017). Overview of the 5th author profiling task at pan 2017: Gender and language variety identification in twitter. *Working Notes Papers of the CLEF*.

Resnik, P. (1999). Semantic Similarity in a Taxonomy: An Information-based Measure and Its Application to Problems of Ambiguity in Natural Language. *J. Artif. Int. Res.*, *11*(1), 95–130. Retrieved June 13, 2019, from http://dl.acm.org/citation.cfm?id=3013545.3013547

Rohde, D. L. T., Gonnerman, L. M., & Plaut, D. C. (2006). An improved model of semantic similarity based on lexical co-occurence. COMMUNICATIONS OF THE ACM. Retrieved June 13, 2019, from http://citeseerx.ist.psu.edu/viewdoc/versions?doi=10.1.1.131.9401

Rubenstein, H., & Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communications of the ACM*, *8*(10), 627–633. DOI: 10.1145/365628.365657

Sahlgren, M. (2008). The distributional hypothesis. *Italian Journal of Disability Studies*, *20*, 33–53.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, *24*(5), 513–523. DOI: 10.1016/0306-4573(88)90021-0

Schütze, H. (1993). Word Space, In *Advances in Neural Information Processing Systems 5, [NIPS Conference]*, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. Retrieved June 13, 2019, from http://dl.acm.org/citation.cfm?id=645753.758140

Schütze, H. (1998). Automatic Word Sense Discrimination. *Comput. Linguist.*, *24*(1), 97–123. Retrieved June 13, 2019, from http://dl.acm.org/citation.cfm?id=972719.972724

Settles, B. (2005). ABNER: An open source tool for automatically tagging genes, proteins and other entity names in text. *Bioinformatics*, *21*(14), 3191–3192. DOI: 10.1093/bioinformatics/bti475

Speer, R., Chin, J., & Havasi, C. (2017). ConceptNet 5.5: An Open Multilingual Graph of General Knowledge, 4444–4451. http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14972

Sutton, C., & McCallum, A. (2012). An Introduction to Conditional Random Fields. *Foundations and Trends® in Machine Learning*, *4*(4), 267–373. DOI: 10.1561/2200000013

Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., & Su, Z. (2008). ArnetMiner: Extraction and Mining of Academic Social Networks, In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Las Vegas, Nevada, USA, ACM. DOI: 10.1145/1401890.1402008

Tateisi, Y., Ohta, T., Miyao, Y., Pyysalo, S., & Aizawa, A. (2016). Typed Entity and Relation Annotation on Computer Science Papers, 3836–3843. http://www.lrec-conf.org/proceedings/lrec2016/pdf/784_Paper.pdf

Tomokiyo, T., & Hurst, M. (2003). A Language Model Approach to Keyphrase Extraction, In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*. http://aclweb.org/anthology/W03-1805

Turney, P. D. [P. D.], & Pantel, P. (2010). From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, *37*, 141–188. DOI: 10.1613/jair.2934

Turney, P. (1997). Extraction of keyphrases from text: Evaluation of four algorithms, National Research Council. *Institute for Information Technology, Technical Report ERB-1051.*

Turney, P. D. [Peter D]. (2000). Learning Algorithms for Keyphrase Extraction. *Information Retrieval*, *2*(4), 303–336. DOI: 10.1023/A:1009976227802

Wartena, C., & Brussee, R. (2008). Topic Detection by Clustering Keywords, In *2008 19th International Conference on Database and Expert Systems Applications*, IEEE. DOI: 10.1109/DEXA.2008.120

Weeds, J. (2003). Measures and Applications of Lexical Distributional Similarity.

Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., & Nevill-Manning, C. G. (1999). KEA: Practical Automatic Keyphrase Extraction, In *Proceedings of the Fourth ACM Conference on Digital Libraries*, Berkeley, California, USA, ACM. DOI: 10.1145/313238.313437

Wolfram, D. (2016). Bibliometrics, Information Retrieval and Natural Language Processing: Natural Synergies to Support Digital Library Research, In *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL)*. Retrieved February 18, 2019, from http://aclweb.org/anthology/W16-1501

You, W., Fontaine, D., & Barthès, J.-P. (2013). An automatic keyphrase extraction system for scientific documents. *Knowledge and Information Systems*, *34*(3), 691–724. DOI: 10.1007/s10115-012-0480-2

Zhang, C., Wang, H., Liu, Y., Wu, D., Liao, Y., & Wang, B. (2008). Automatic Keyword Extraction from Documents Using Conditional Random Fields. *Journal of Computational Information Systems*, *4*(3), 1169–1180. http://eprints.rclis.org/12305/1/Automatic_Keyword_Extraction_from_Documents_Using_Conditional_Random_Fields.pdf

Zhang, Y., Zincir-Heywood, N., & Milios, E. (2004). Term-Based Clustering and Summarization of Web Page Collections (A. Y. Tawfik & S. D. Goodwin, Eds.). In A. Y. Tawfik & S. D. Goodwin (Eds.), *Advances in Artificial Intelligence*, Springer Berlin Heidelberg.