



**HAL**  
open science

# Enabling Dynamic Interactions between Natural Language and Structured Knowledge Bases

Hady Elsahar

► **To cite this version:**

Hady Elsahar. Enabling Dynamic Interactions between Natural Language and Structured Knowledge Bases. Neural and Evolutionary Computing [cs.NE]. Université de Lyon, 2019. English. NNT : 2019LYSES022 . tel-04675310

**HAL Id: tel-04675310**

**<https://theses.hal.science/tel-04675310v1>**

Submitted on 22 Aug 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



N° d'ordre NNT : 2019LYSES022

**THESE de DOCTORAT DE L'UNIVERSITE DE LYON**

opérée au sein de

**Université Jean Monnet**

**Ecole Doctorale ED SIS 488**

**(Ecole Doctorale Science, Ingénierie et Santé)**

**Spécialité de doctorat:**

Informatique

Soutenue publiquement le 05/07/2019, par

**Hady Elsahar**

**Enabling Dynamic Interactions between Natural  
Language and Structured Knowledge Bases**

Devant le jury composé de :

Marie-Francine MOENS	Professeure	KU Leuven	Rapporteure
Roberto NAVIGLI	Professeur	Sapienza Università di Roma	Rapporteur
Pascal PONCELET	Professeur	L'Université de Montpellier	Examineur
Laure SOULIER	Maître de conférences	Sorbonne Université	Examinatrice
Frederique LAFOREST	Professeur	INSA Lyon	Directrice de thèse
Christophe GRAVIER	Maître de conférences	Université Jean Monnet	Co-directeur de thèse

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Knowledge Bases and Natural Language interactions . . . . .	1
1.1.1	Motivation . . . . .	3
1.2	Research Questions and challenges . . . . .	5
1.3	Thesis Contributions . . . . .	9
1.3.1	R1 Relation Discovery . . . . .	9
1.3.2	R2 Limitation of Training Data . . . . .	10
1.3.3	R3 NLG for Automatic generation of entity descriptions . . . . .	11
<b>2</b>	<b>Background</b>	<b>13</b>
2.1	Introduction . . . . .	13
2.2	Dependency Parsing . . . . .	13
2.2.1	Dependency Grammar . . . . .	14
2.2.2	Transition-based Dependency Parsing . . . . .	15
2.3	Word Vector Representations . . . . .	15
2.4	Sequence To Sequence Models (seq2seq) . . . . .	16
2.4.1	Seq2Seq with Attention Models . . . . .	17
2.4.2	Training Sequence to Sequence models . . . . .	18
2.4.3	Sequence to Sequence models at inference time . . . . .	19
2.4.4	Copy Actions . . . . .	20
2.4.5	Evaluation of Natural Language Generation . . . . .	21
2.5	Tasks Related to Knowledge Graphs Relationships . . . . .	24
2.5.1	Open Relation (Information) Extraction . . . . .	24
2.5.2	Relation Extraction (classification) . . . . .	25

<i>CONTENTS</i>	3
2.5.3 Link Prediction or Knowledge Base Completion . . . . .	27
2.5.4 Relation Discovery . . . . .	27
<b>3 Limitations of Knowledge Bases</b>	<b>30</b>
3.1 Unsupervised Open Relation Extraction . . . . .	30
3.1.1 Proposed Method . . . . .	31
3.1.2 Evaluation . . . . .	33
3.1.3 Conclusion . . . . .	34
3.2 High Recall Open IE for Relation Discovery . . . . .	35
3.2.1 Our Approach . . . . .	37
3.2.2 Experiments and Evaluation . . . . .	39
3.2.3 Conclusion . . . . .	42
<b>4 Limitations of Training Data</b>	<b>43</b>
4.1 T-REx: A Large Scale Alignment of Natural Language with Knowledge Base Triples . . . . .	43
4.1.1 Related Work . . . . .	44
4.1.2 <i>T-REx</i> Creation . . . . .	46
4.1.3 <i>T-REx</i> Dataset . . . . .	48
4.1.4 Evaluation . . . . .	49
4.1.5 Error Analysis . . . . .	51
4.1.6 Conclusion . . . . .	52
4.2 Zero-Shot Question Generation from Knowledge Graphs for Unseen Predicates and Entity Types . . . . .	54
4.2.1 Related Work . . . . .	55
4.2.2 A Model for Zero-Shot QG . . . . .	57
4.2.3 Textual Context Encoder . . . . .	58
4.2.4 Textual contexts dataset . . . . .	61
4.2.5 Experiments . . . . .	63
4.2.6 Results & Discussion . . . . .	68
4.2.7 Conclusion . . . . .	69

<i>CONTENTS</i>	4
<b>5 Limitations of Answer Display</b>	<b>71</b>
5.1 Learning to Generate Wikipedia Summaries for Underserved Languages from Wikidata . . . . .	71
5.1.1 Model . . . . .	73
5.1.2 Implementation and Training Details . . . . .	75
5.1.3 Dataset . . . . .	76
5.1.4 Baselines . . . . .	77
5.1.5 Results and Discussion . . . . .	78
5.1.6 Conclusions . . . . .	80
5.2 Mind the (Language) Gap: Generation of Multilingual Wikipedia Summaries from Wikidata for ArticlePlaceholders . . . . .	81
5.2.1 Related Work . . . . .	82
5.3 Methods . . . . .	84
5.3.1 Training and Automatic Evaluation . . . . .	86
5.3.2 Community Study . . . . .	88
5.3.3 Results and Discussions . . . . .	91
5.3.4 Community Study . . . . .	92
5.3.5 Conclusion . . . . .	94
<b>6 Conclusions and Future work</b>	<b>95</b>
6.1 Summary and Conclusions . . . . .	95
6.2 Future Directions . . . . .	98
6.2.1 Data Augmentation Using Self-training . . . . .	98
6.2.2 Data Augmentation for NLG Through Back Translation . . . . .	99
6.2.3 Handling Domain Shift . . . . .	99
<b>Bibliography</b>	<b>100</b>

# List of Tables

2.1	Selected dependency relations from the universal dependency grammar	14
2.2	Table summarizing various tasks dealing with text and knowledge bases.	28
3.1	Comparison between different sentence representations when used as features for clustering.	34
3.2	Pairwise $F_1$ (%) scores of different models on the test set of the NYT-FB dataset.	34
3.3	Example sentences where all OpenIE systems failed to extract target relations, and their corresponding Predicate-Centric extractions.	39
3.4	pairwise $F_1$ scores using word embeddings and sparse features (Em-Ft), after re-weighting word embeddings (wEm-Ft), after doing feature reduction (wEm-Ft-PCA), and combining all features (ALL).	42
4.1	: Statistics over existing alignments from previous work.	45
4.2	Comparison between different extractions of three alignment schemes for a sample paragraph of two sentences. The detected properties in the paragraph are put between square brackets. Wrong alignments are in italic.	48
4.3	Number of alignments in different datasets	49
4.4	Accuracy of each alignment methodology in T-REx	51
4.5	Accuracy of top properties for each annotation methodology in T-REx	51
4.6	Causes of error in alignments	52
4.7	An annotated example of part-of-speech copy actions from several input textual contexts (C1, C2, C3), the words or placeholders in bold are copied in the generated question	60

4.8	Table showing an example of textual contexts extracted for freebase predicates . . . . .	63
4.9	Dataset statistics across 10 folds for each experiment . . . . .	64
4.10	Evaluation results of our model and all other baselines for the unseen predicate evaluation setup . . . . .	65
4.11	Automatic evaluation of our model against selected baselines for unseen sub-types and obj-types . . . . .	66
4.12	results of Human evaluation on % of predicates identified and naturalness 0-5 . . . . .	68
4.13	Examples of generated questions from different systems in comparison .	70
5.1	Recent page statistics and number of unique words (vocab. size) of Esperanto, Arabic and English Wikipedias. . . . .	73
5.2	Training example: a set of triples about <i>Florida</i> . Subsequently, our system summarises the input set in the form of text. The vocabulary extended summary is the one on which we train our model. . . . .	73
5.3	Dataset statistics in Arabic and Esperanto. . . . .	77
5.4	Automatic evaluation of our model against all other baselines using BLEU 1-4, ROUGE and METEOR for both Arabic and Esperanto Validation and Test set . . . . .	78
5.5	Participation Numbers: Total number of Participants ( $P$ ), Total number of Sentences ( $S$ ), Number of $P$ that evaluated at least 50% of $S$ , and average number of $S$ evaluated per $P$ . . . . .	88
5.6	Automatic evaluation of our model against all other baselines using BLEU1-4, ROUGE and METEOR for both Arabic and Esperanto Validation and Test set . . . . .	90
5.7	Results for fluency and appropriateness . . . . .	92
5.8	Percentage of summaries in each category of reuse in Arabic and Esperanto. An example is provided for each category containing a generated summary (top) and after it is was edited (bottom). Solid lines represent reused tiles, while dashed lines represent overlapping sub-sequences smaller than $mml$ and not contributing to the $gstscore$ calculation. . . . .	93

# List of Figures

1.1	A web page containing information about the entity Earth (id Q2) from the Wikidata knowledge base. . . . .	2
1.2	Increase of number of websites (unique hostnames) in the past two decades	4
1.3	Increase in Number of entities on Wikidata knowledge base in the past years . . . . .	5
1.4	Increase in the number of Wikidata properties . . . . .	6
1.5	Snippet answer from Google Assistant containing descriptions about Soweto extracted from Wikipedia . . . . .	9
2.1	The dependency graph representation of an example sentence in Enhanced English Universal Dependencies . . . . .	14
2.2	Basic sequence to sequence model using neural networks from Sutskever et al. [164] . . . . .	16
2.3	Figure from [186] showing the first attention mechanism developed for image captioning. In order for the decoder to output a distribution over the of vocabulary at each decoding time step, it attends to certain parts of the input image . . . . .	17
2.4	Figure from [5] showing implementation architecture of Attention mechanism for Seq2Seq models, next to it a visualization of output tokens attention weights to each input token for the task of Neural Machine Translation. . . . .	18
2.5	An example from Luong et al. [102] showing one proposed way to model copy actions using special tokens from the input and output vocabulary. . . . .	21



2.6	The pointer generator model from [150] showing the output probability distribution over output words is calculated using the regular generation mechanism with probability $p_{gen}$ or through copying mechanism using the attention weights from the attention mechanism with probability $1 - p_{gen}$ . . . . .	22
2.7	Several OpenIE outputs for the same sentence "If he wins five key states, Republican candidate Mitt Romney will be elected President in 2008." from [125] . . . . .	25
3.1	Sentences containing textual <u>relations</u> between <i>named entities</i> . . . . .	30
3.2	Our Proposed system for relation discovery and clustering . . . . .	32
3.3	Examples of textual representations mentioning the predicate "official currency". . . . .	36
3.4	Distribution of sentences in the NYT corpus (A), which have: (B) at least 1 entity mention, (C) at least 1 entity and a predicate attached to it, (D) at least 2 entities mentions, (E) at least 2 entities and a relation in between in Freebase. . . . .	37
3.5	Maximum recall of top Open IE systems and their corresponding precisions in comparison with our approach <b>RelDiscovery</b> on [159] evaluation dataset. . . . .	40
3.6	Precision and recall curve of our relation discovery method <b>RelDiscovery</b> with different OpenIE systems. . . . .	41
4.1	Overview of the alignment pipeline and its components . . . . .	45
4.2	Distribution of the number of alignments created for each predicate . . . . .	50
4.3	The proposed model for Question Generation. The model consists of a single fact encoder and $n$ textual context encoders, each consists of a separate GRU. At each time step $t$ , two attention vectors generated from the two attention modules are fed to the decoder to generate the next word in the output question. . . . .	56
5.1	Model Overview . . . . .	75
5.2	A box plot showing the distribution of BLEU 4 scores of all systems for each category of generated summaries. . . . .	79

5.3 The triple encoder computes a vector representation for each one of the three input triples from the ArticlePlaceholder,  $h_{f_1}$ ,  $h_{f_2}$  and  $h_{f_3}$ . Subsequently, the decoder is initialized using the concatenation of the three vectors,  $[h_{f_1}; h_{f_2}; h_{f_3}]$ . The purple boxes represent the tokens of the generated summary. Each summary starts and ends with the respective start-of-summary `<start>` and end-of-summary `<end>` tokens. . . . . 85

# Enabling Dynamic Interactions between Natural Language and Structured Knowledge Bases

Hady Elsahar

July 4, 2019

# Chapter 1

## Introduction

### 1.1 Knowledge Bases and Natural Language interactions

Natural language has always been the most straight forward way of communication and documenting information in terms of books and messages. Alternatively and intuitively with the invention of numbers, the humanity discovered the strength of statistics and structured information, in which information are displayed in rows and columns <sup>1</sup> or possibly more complex structures that allow better visual comprehension and facilitate operations over information of the same type. The interaction between these two types of information representation was a very natural thing to happen as both are seen to be complementary to each other. Over the history we have seen tables in papyrus, hand-written notes, print media <sup>2</sup> and software. Yet still, natural language was used to add context to structured information, such as table descriptions for examples.

Since then, the use of structured information in our daily life has grown more and more. Unsurprisingly since it has shown to be a very compact and efficient way to storing and querying information, either by human or even using machines in the modern era. Fast forward to the last decade, knowledge bases in their modern form have

---

<sup>1</sup>An ancient Egyptian slab stela painted on lime stone (dated 2590-2565 BC) depicting a table to count offerings given to Neferetiabet a possible daughter of Pharaoh Khufu. <https://commons.wikimedia.org/wiki/File:Neferetiabet.jpg>

<sup>2</sup>Mendeleev's periodic table in - Zeitschrift für Chemie (1869, pages 405–6) [https://commons.wikimedia.org/wiki/File:Mendeleev%27s\\_periodic\\_table\\_\(1869\).svg](https://commons.wikimedia.org/wiki/File:Mendeleev%27s_periodic_table_(1869).svg)

emerged from standard databases and have been widely used in large amount of applications. Knowledge bases such as Wikidata [176] and Yago [162] consist of information that represent facts about the world either in the general domain or on a specific domain. These knowledge bases are usually paired with inference engines that can perform reasoning over these facts and use logical rules to deduce new facts or to detect inconsistencies.

**Earth** (Q2) [edit](#)

third planet from the Sun in the Solar System [edit](#)

[Blue Planet](#) | [Terra Mater](#) | [Terra](#) | [Planet Earth](#) | [Tellus](#) | [Sol III](#) | [Sol 3](#) | [Sol d](#) | [♁](#) | [♁](#) | [♁](#) | [Blue Marble](#) | [🌍](#) | [The Earth](#) | [♁](#) | [Gaia](#) | [The world](#) | [Globe](#) | [The Blue Gem](#)

[In more languages](#)

### Statements

instance of	<a href="#">inner planet</a> ▼ 0 references	<a href="#">edit</a> <a href="#">+ add reference</a> <a href="#">+ add value</a>
part of	<a href="#">Earth-Moon system</a> ▼ 0 references	<a href="#">edit</a> <a href="#">+ add reference</a>
	<a href="#">Solar System</a> ▼ 0 references	<a href="#">edit</a> <a href="#">+ add reference</a>
	<a href="#">Milky Way</a> ▼ 0 references	<a href="#">edit</a> <a href="#">+ add reference</a> <a href="#">+ add value</a>


image	 <a href="#">The Earth seen from Apollo 17.jpg</a> 3,000 × 3,002; 6.21 MB	<a href="#">edit</a>
-------	--	----------------------

Figure 1.1: A web page containing information about the entity Earth (id Q2) from the Wikidata knowledge base.

Interactions between Knowledge Bases and Natural Language have always motivated a lot of applications. Interactions between Knowledge Bases and Natural Language have always motivated a lot of applications. In the course of this thesis we will focus on one major application which is Question Answering (QA). Modern QA systems operate over structured information and specifically knowledge bases either partially or totally. The past few years, QA systems have been a major part in developing personal assistants such as iPhone Siri, Amazon Alexa and Google Assistant, which has been an ongoing industrial race between big players in the market. Question Answering systems in personal assistants are usually equipped partially with a structured knowledge base from which they try to find answers, entities and information to allow interactions with the user in a pure natural language.

Another form of interaction between structured knowledge and natural language is natural language generation (NLG) from structured information [38, 43, 65]. Commercial NLG is growing rapidly and has been successful lately since its early applications of weather forecast generation [57] and robo journalism [117]. Companies such as Arria NLG<sup>3</sup>, Automated insights<sup>4</sup>, Narrative Science<sup>5</sup> and Ax Semantics<sup>6</sup> started providing services in the market that generate natural language articles and summaries from structured information for many applications such as marketing and sales analytics, sports articles, e-commerce, tourism and financial reporting.

### 1.1.1 Motivation

The amount of information and data in the world is in a continuous rapid growth. This growth is manifested in almost every aspect of the web content. For example as shown in Figure 1.2 the number of registered new domains on the world wide web has doubled only in the last two years reaching over 1.9 billion registered domains as of 2018<sup>7</sup>.

Not only in terms of count but also the size of the web content, the average size of each web page has grown to 3.5 MB in 2018; this is double of what it was 4 years ago.<sup>8</sup>

---

<sup>3</sup><https://www.arria.com/>

<sup>4</sup><https://automatedinsights.com/>

<sup>5</sup><https://narrativescience.com/>

<sup>6</sup><https://www.ax-semantics.com/en/home.html>

<sup>7</sup>source <http://www.internetlivestats.com/total-number-of-websites/>

<sup>8</sup>source <https://speedcurve.com/blog/web-performance-page-bloat/>

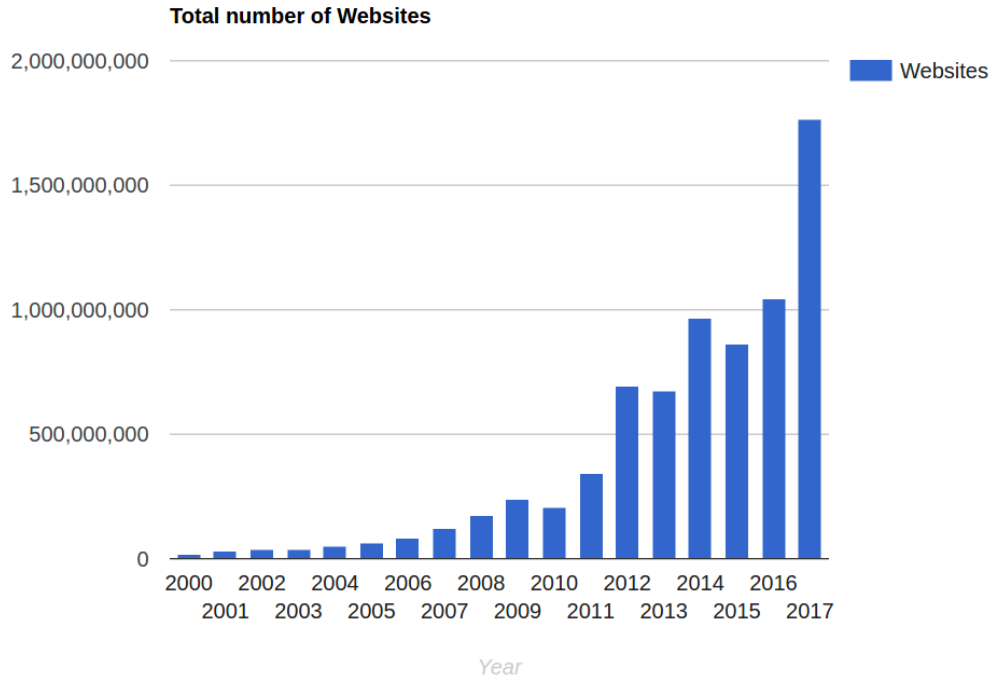


Figure 1.2: Increase of number of websites (unique hostnames) in the past two decades

This fast paced growth is not only manifested in unstructured web pages but also in the size of structured information on the web. Similar growth patterns were also seen in structured knowledge bases, for example the number of entities on the Wikidata Knowledge base is doubled from 26 million to 52 million in the past two years <sup>9</sup>.

On one had his flood of information raises many potentials such as more coverage and representation of new domains and languages. However, on the other hand these potentials are accompanied with many research challenges, when it comes to applications that are concerned with interactions between structured and non-structured information. Many of the existing applications can rapidly become inapplicable for these new pieces of information being published. For example, in terms of data representation, a schema of a specific knowledge base might not be able to represent a new entity type or a new relation that was published recently.

This problem becomes more critical with the recent wide usage of Black Box end-to-end data driven approaches, in which a limited training dataset collected from a specific time

<sup>9</sup>source <https://grafana.wikimedia.org/d/000000167/wikidata-datamodel>

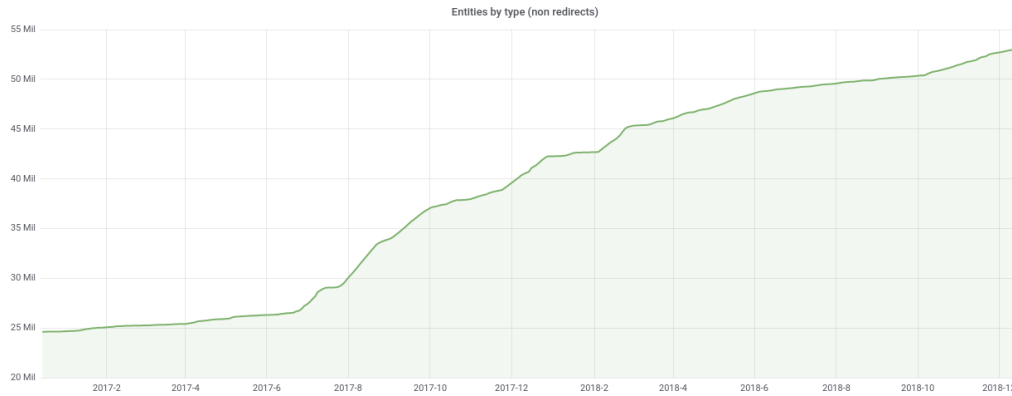


Figure 1.3: Increase in Number of entities on Wikidata knowledge base in the past years

span is used to train models for several tasks. Those models can face several challenges to generalize to new pieces of information that do not similarly match to any of the examples in the training datasets.

The synergies between KB and NL are therefore of paramount importance for cutting-edge QA systems. This thesis contributes to foster these synergies by tackling the major challenges presented in the next section.

## 1.2 Research Questions and challenges

### R1: Limitation of Knowledge Bases

Modern knowledge bases store information in the form of triples (S,P,O): subject, predicate, object. Each triple represents a semantic directional relationship between the two entities of the triples. Each of the entities and the predicate in a triple is represented by a unique id in this knowledge base. For example the statement "London is the capital of United Kingdom" is represented in Wikidata Knowledge base as  $(Q84, P1376, Q145)$  where  $Q84$  and  $Q145$  are the unique identifiers of the entities "London" and "United Kingdom" respectively, while  $P137$  is the unique identifier of the predicate "capital of". Existing available and proprietary knowledge bases are numerous. Some of those are built purely using automatic techniques such as DBpedia [94], BabelNet [121], Knowledge Vault [34] and Yago [162], or using crowdsourcing of human volunteers and automated bots such as Wikidata [176]. What is common between most of those knowledge bases is that they mostly rely on a defined ontology. Knowledge base ontologies define



a set of concepts (entity types) that could be contained in a knowledge base as well as a set of predicates that is allowed to be attached to each of those concepts. Afterwards a knowledge base is being filled with information following those ontologies, for example triples describing an entity of type "city" should only contain a set of predefined predicates such as "population", "area", "located in" and not contain relations describing predicates such as "father of" or "born in" which are not applicable to entities of this type. Ontologies allow operations over knowledge bases triples such as inference and fact verification. However a rigid ontology will suffer to represent all world information, since many entities with possibly new class types appear over time. those new types will have to be attached to their corresponding relevant predicates that describe them, or possibly new predicates should be created in the schema. This lead some projects to iteratively adapt their ontologies with new class types and new predicates. For example as shown in Figure 1.4 the number of predicates in Wikidata has almost doubled from 2.9K to 5.6K only in the last two years <sup>10</sup>.

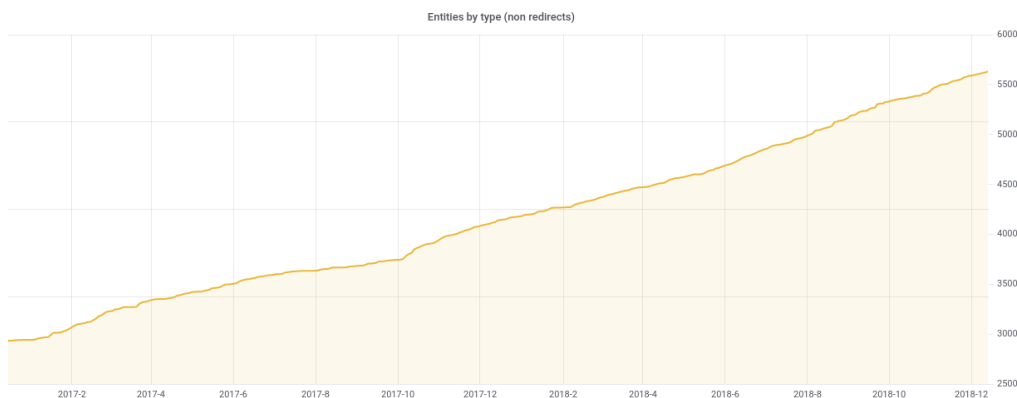


Figure 1.4: Increase in the number of Wikidata properties

This adaptation in the ontology is very natural and expected to accommodate the rapid increase in the large number of entities and class types being created everyday on knowledge bases. Discovery, addition or removal of such predicates and class types is a tedious task and usually includes lots of discussion from the community or experts <sup>11</sup>. Even for projects supported by a large community of volunteers such as Wikidata, with

<sup>10</sup><https://grafana.wikimedia.org/d/000000167/wikidata-datamodel>

<sup>11</sup>[https://www.wikidata.org/wiki/Wikidata:Property\\_proposal/Generic](https://www.wikidata.org/wiki/Wikidata:Property_proposal/Generic)

the fast pace of new information being available on the web everyday, this task can easily become intractable.

All this leads to our first research question:

**R1:** How can we develop techniques that help ontology designers to identify and discover new emerging predicates and class types?

## **R2: Limitation of Training Data**

Several tasks in NLP are concerned with reducing the gap between Natural Language and structured Knowledge Bases (KB), such as Relation Extraction and KB Population [112, 77], Natural Language Generation from knowledge bases [92, 174] and Question Answering over knowledge bases [188]. In the recent years supervised end-to-end models have become the de facto methods for those tasks. Those models rely on training datasets containing alignments between sentences, paragraphs or questions in free text and their corresponding KB triples. With the rapid increase of information and the rapid change in current schemas of knowledge bases, existing training datasets can easily become limited in size and coverage. Models trained on such examples cannot handle examples at test time that contain predicates and entity types which are not covered in the training datasets. For example a Question Answering model trained on a QA dataset such as Simple Questions [15]<sup>12</sup> which was published few years ago will have a hard time dealing with questions such as: *"Which professional gamer was named the most valuable player in the overwatch e-sports league?"*. This is mainly due to the existence of emerging class types such as "e-sports league" and "professional gamer"; these class types were not as popular as now and are not contained in the training examples used to train the QA model. One would assume the solution to this problem is to continuously keep updating the training datasets with new examples and retrain corresponding models. While this might be a possible solution, since this overhead is anticipated and models in production have always to undergo maintenance and adaptation. However this still can be challenging, as many of these datasets are created manually using crowdsourcing, having a continuous process of manually annotating new examples can be very expensive to sustain, since such models expect a significant amount of examples for each new predicate for learning. Automatic ways of creating datasets are possible

---

<sup>12</sup>A dataset containing natural language questions paired with their answers from a specific knowledge base

solutions to overcome this and achieve continuously adapting learning resources for KB and natural language interactions. However, for the time being, available research solutions for automatically creating such datasets suffer from lack of reproducibility and barely report any quality of the automatically created datasets.

Therefore, the second research question tackled in this thesis is :

**R2:** Can we building frameworks and techniques for continuously evolving resources for training models concerned with KB and natural language interactions?

### **R3: Limitations of Answer Display**

Knowledge bases rely on underlying representations that might not be ideal to display to users as is. Therefore many projects display KB information to the users in terms of visuals, extended with a natural language description of the target information. This can be seen in answer displays of many QA systems results as in Google Assistant Figure 1.5, or projects for visualizing knowledge bases such as Wikimedia Reasonator<sup>13</sup> These textual descriptions of entities are usually embedded from sources on the web such as Wikipedia. For popular KB entities, there is always a corresponding Wikipedia article or a web resource describing it, in which textual information can be extracted in most of the well-supported languages on the web. However, with the fast growth of information on the web, a huge number of new entities are being created daily in knowledge bases, for example in the past year there were more than 10 million newly created entities on Wikidata (37K per day) see figure 1.3. Those emerging new entities do not have enough content in natural language on the web about them to extract textual descriptions about them. This problem becomes even more prevalent for under-served languages, where content on the web becomes even harder to find.

Techniques for Automatic Natural Language Generation from structured data can become handy to deal with such problem. This motivates the third research question in this thesis:

**R3:** How NLG can help in automatically generating descriptions of emerging Knowledge base entities in a fully dynamic way?

<sup>13</sup><https://tools.wmflabs.org/reasonator/?q=Q42>

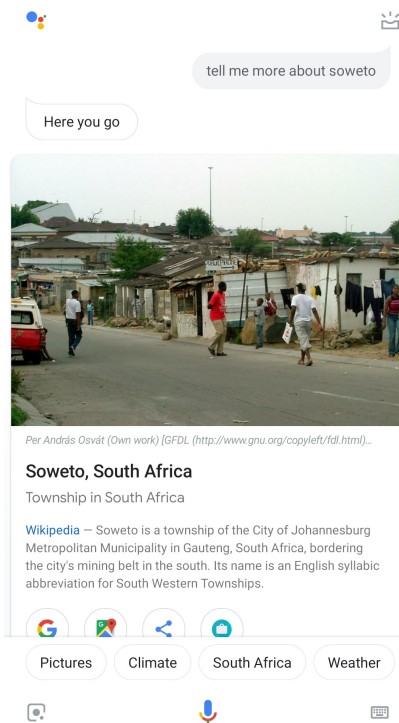


Figure 1.5: Snippet answer from Google Assistant containing descriptions about Soweto extracted from Wikipedia

## 1.3 Thesis Contributions

For each of the challenges discussed in the previous section, I will discuss the research that has been done and the contributions of our investigations.

### 1.3.1 R1 Relation Discovery

As discussed in the previous section, one of the challenges caused by the rapid increase in online information is the need of adapting ontologies of existing knowledge bases. To tackle this problem, we looked into the problem of Relation Discovery. Relation Discovery identifies predicates (relation types) from a text corpus relying on the co-occurrence of two named entities in the same sentence.

In chapter 3 we propose two contributions tackling this problem. Our first contribution (see our publication in ESWC'17 [45]) is a new relation discovery system that is able to extract and cluster relation mentions in an unsupervised way in a large unstructured

corpus. It relies on the assumption that any sentence that mentions two named entities might exhibit a relation in between. Afterwards it builds sentence representations based on the words mentioned in the sentence as well as types of the mentioned named entities. For the latter, to solve the problem of representation sparsity we explore several methodologies in order to compress the sentence representation into a more dense one. Next, we cluster these dense sentence representations such that semantically equivalent relations are mapped to the same cluster. Our method yields the state-of-the-art results in relation discovery.

Although our proposed method in the first contribution of chapter 3 achieves the state-of-the-art in relation discovery. It suffers – like similar methods from the literature – from a very low recall value. This is mainly because filtering the large text corpora to only sentences with a mention of two entities to detect relations in between, this leads to a set of sentences that only represent a small fraction of all relation mentions in practice. In order to alleviate this problem we propose in the second part of chapter 3 a high recall approach for predicate extraction which enables covering up to 16 times more sentences in a large corpus. We compare our approach against state-of-the-art Open Information extraction (OpenIE) systems and we show that our proposed approach achieves 28% improvement over the highest recall OpenIE system. This means that our approach is capable of extracting larger number of relation mentions in a corpus than traditional methods that rely on sentences that only include a mentioned of two named entities.

### 1.3.2 R2 Limitation of Training Data

In order to alleviate the need of continuous data annotation, many techniques in the literature tried to alleviate this by automatically generating training datasets, for example as in distance supervision methods for relation extraction [113] or using language generation to enhance the quality of Simple Question Answering [152].

The main contributions of this thesis considering automatic generation of datasets is split into two main contributions. In the first part of chapter 4, I present T-REx [47] a framework for creating automatic alignments between knowledge base triples and sentences from free text using a distant supervision assumption. In this work we tackle many of the issues in existing datasets of similar kind, such as limited size, limited coverage, unreported quality and lack of reproducibility over new text corpora or new knowledge bases. The result of running the T-REx framework over the Wikipedia abstracts dataset

and Wikidata knowledge base has yielded the largest available alignments in the literature between KB triples and free text. We additionally perform an extensive error analysis using crowd-sourcing to report the quality of those alignments. With regard to reproducibility T-REx framework has proven to be easy to run on new KBs and text corpora, many of next chapters of the thesis has relied on the T-REx framework to acquire training datasets such as in multilingual KB triples summarization in chapter 5, aligning questions with KB triples in chapter 4 or alignment of DBpedia with Wikipedia text as in [174] which is not included in this thesis.

The second contribution in Chapter 4 is a new technique for zero-shot question generation from knowledge graphs. Question generation has been used before as a technique for data augmentation for enhance the performance of question answering systems [152, 33, 85]. However none of the previous work has tackled the problem of dynamically augmenting existing QA datasets with questions describing new relation types and new entity types that haven't been seen during training time. This means that current Question Answering systems after being augmented with extra generated datasets will still face problems answering questions about emerging relations and entity types. In order to alleviate this problem we propose a new technique for generating training datasets for QA systems under a zero-shot setup [46]. We enrich traditional sequence to sequence models that have been used before for question generation with textual inputs aligned to the input triple. This provides the potential of generalizing to unseen predicates and entity types by providing to the seq2seq model sufficient input vocabulary to express them. We equip this technique with a novel copy actions based on linguistic features which enhances the generalizability to new vocabulary expressing new relations and entity types.

### 1.3.3 R3 NLG for Automatic generation of entity descriptions

As discussed in the previous section, textual descriptions of knowledge base entities are very crucial corner stone of displaying answers to users in any Question Answering system. However due to the large number of entities that emerge everyday to structured knowledge bases, textual descriptions of such entites might not be available neither on Wikipedia or the internet, specially in a multilingual setting. We tackle this problem by using techniques of natural language generation from structured data. The main contributions of this thesis is manifested in two publications [82, 83] as shown in Chapter 5. In

this work we explore the viability of automatically generating textual summaries for entities in knowledge bases relying on automatically generated training datasets that aligns knowledge base triples with entity descriptions from free text, this relying on techniques from Chapter 4. As an extreme case to manifest the lack of available information for such new emerging entities and hence the lack of available training data, we choose to apply our approach to generate summaries in underserved languages from which we choose Arabic and Esperanto as usecases.

Following the recent success of recurrent models for natural language generation, we propose a neural network architecture for natural language generation from structured information. Our proposed architecture takes a set of Knowledge base triples describing the target entity as an input, and outputs a textual summary for this entity one word each time in the target language. In order to overcome the potential shortage of training data, we support our proposed model with copy actions based on surface forms of entities and entity types, those copy actions delexicalize the output with a set of placeholders which reduce the size of the output vocabulary and hence make the model more data efficient. We rely on two methodologies for evaluation: First through an automatic evaluation by measuring how close the generated summaries by the proposed model to the actual reference summaries in Wikipedia, we compare against two other baselines of different natures: a unconditional language model, and a template-based approach. Second by assessing the usefulness of the generated summaries using a qualitative evaluation involving readers and editors of underserved Wikipedias in which we ask them to evaluate the generated summaries in terms of their fluency, appropriateness for Wikipedia, and the percentage of the reuse if they were to write an introductory paragraph about the same entity in hand.

## **Chapter 2**

# **Background**

### **2.1 Introduction**

This chapter contains background and related works to the contributions discussed in this thesis. It is not a related work chapter per se, but on the contrary serves as a reference for the many technical and architectural concepts driving modern NLP systems. More in-depth related work is included in each chapter. It starts by explaining some core NLP components that has been widely utilized in systems that manages interactions between structured knowledge bases and unstructured free text such as Dependency parsing in Section 2.2, Word Embeddings Section 2.3, Neural Sequence to Sequence models 2.4 which is a core recent technique for neural language generation. Finally it ends by listing down in Section 2.5 several tasks that is meant with extraction of structured information from knowledge bases.

### **2.2 Dependency Parsing**

Since the start of dependency semantic representations [29], Dependency Parsing has been crucial to a wide range of shallow natural language understanding tasks such as recognizing textual entailment [136], relation extraction [114], open domain relation extraction (OpenIE) [107]. Dependency representations were beneficial to such applications since they provide information about predicate-argument structure which are not



directly available from other parsing structures such as constituency parsing for example.

### 2.2.1 Dependency Grammar

Dependency grammars are based on the linguistic grammatical relations which are used to label functional relationships between constituents in a clause. In a clause, for each constituent, there is a head word which can be considered as the root of this constituent and the rest of the words in the constituents are dependent either directly or indirectly to this head word. One of the most widely used grammatical schemes for dependency relations are The Universal Dependencies [127]. This initiative provides a set of dependency relations that are cross-linguistically motivated, as well as their corresponding annotated treebanks in more than 60 languages. Table 2.1 shows some examples of the Dependency relations from the Universal Dependency grammar and figure 2.1 shows an example of a parsed sentence.

Dependency Relation	Description
NSUBJ	Nominal subject
DOBJ	Direct object
IOBJ	Indirect object
CCOMP	Clausal compliment
NMOD	Nominal modifier
AMOD	Adjectival modifier
NUMMOD	Numeric modifier
APPOS	Appositional modifier
CONJ	Conjunction
CC	Coordinating conjunction
DET	Determiner

Table 2.1: Selected dependency relations from the universal dependency grammar

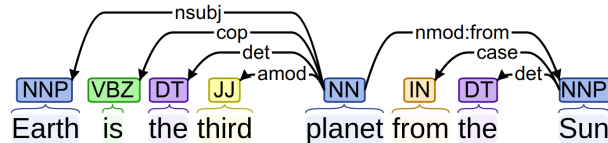


Figure 2.1: The dependency graph representation of an example sentence in Enhanced English Universal Dependencies

## 2.2.2 Transition-based Dependency Parsing

The Transition-based dependency parsing operates basically by reading tokens of a phrase sequentially and combining them incrementally into a parsed structure using a set of decisions at each time step [193, 126]. With the rise of deep learning methods for NLP the most accurate and efficient transition based dependency parsing has been performed using fully connected neural networks [21] or sequential models such as Long Short Term Memory neural networks (LSTM) [41].

## 2.3 Word Vector Representations

Word vector representations represent words in a sentence in a continuous vector space where words with similar meanings are given points that are close in the vector space. As elaborated in [8], methods for learning word representations can be divided into count-based methods in which word representations are built according to how often some word co-occurs with other words in a corpus, and predictive methods which build representations of words by trying to predict a word given its context. Both methods define the meaning by collocation characteristics [111] of natural language in which a word can be represented by the company it keeps.

### Count-based methods

Count models have such a long and rich history in which most of the algorithms start by building a co-occurrence matrix of each word. The co-occurrence matrix is built by sliding a window of fixed size around each word in a large corpus and calculating the co-occurrence counts of each word with its neighbours. Because of the expected sparsity of this matrix, basic count-based methods don't work that well. Thus different transformations were applied to these raw vectors using different reweighting techniques instead of simple counts such as Point-Wise mutual information (PMI) [17] and TFIDF [158] or dimensionality reduction to find a low rank approximation and reduce sparsity of this co-occurrence matrix such as singular value decomposition [58] which yielded the widely known latent semantic analysis (LSA) [40] technique for document representations, and non-negative matrix factorization [93].

### Predictive Neural Methods

Last few years have shown a success in predictive models for learning word representations. It frames the vector estimation problem as the supervised task of predicting the context given a word (or sometimes the opposite) [11, 110, 131, 168] with which efficient word representations are being learned to maximize the probability of the words in the context. One of the reasons of the success of these approaches is that they can be trained with no manual annotation costs, as training of such models can be done on a large un-annotated text corpora. There are various prediction models. One of the most known is the Continuous Bag of Words Model as one of the models used to generate the infamous Word2Vec embeddings [110]. Pre-trained word representations have become the de facto feature representations for almost all NLP problems surpassing almost all state-of-the-art models that rely on hand-crafted features [27].

## 2.4 Sequence To Sequence Models (seq2seq)

Sequence to Sequence models [23, 164] are models that rely on neural networks to transfer sequences of arbitrary lengths to output sequences of arbitrary lengths. Since their introduction they became the defacto solution for almost every natural language generation task, such as neural machine translation, abstractive summarization, sentence compression, sentence simplification and Question Generation. Early versions of sequence to sequence models [164] (figure 2.2) uses a recurrent neural networks to map the entire input sentence to vector, then uses this representation to compute the probability of the output sequence. The model stops making predictions after outputting the end-of-sentence token  $\langle EOS \rangle$ .

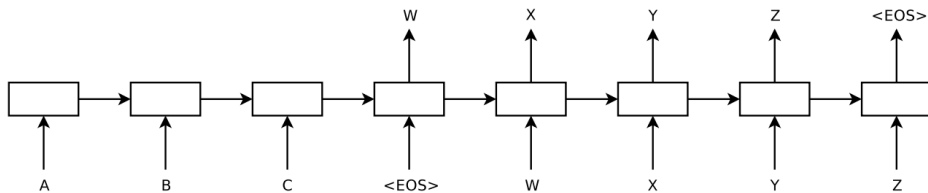


Figure 2.2: Basic sequence to sequence model using neural networks from Sutskever et al. [164]

### 2.4.1 Seq2Seq with Attention Models

Attention Mechanism for seq2seq models is a mechanism which is capable of focusing on special regions of the input while still perceiving the other regions of the inputs with less focus. This is loosely inspired from visual attention mechanism found in humans and were first introduced for image recognition tasks [186] and then later became widely used for various NLP tasks. For a typical NLP task such as Neural Machine Transla-

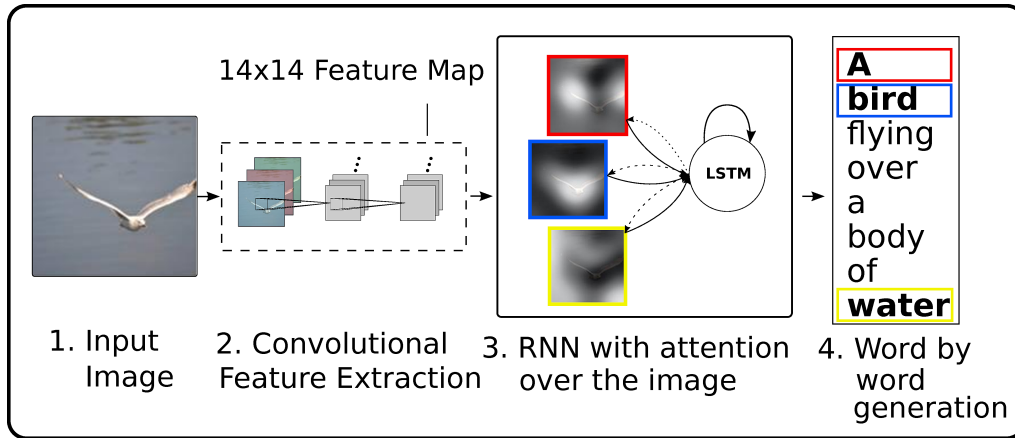


Figure 2.3: Figure from [186] showing the first attention mechanism developed for image captioning. In order for the decoder to output a distribution over the of vocabulary at each decoding time step, it attends to certain parts of the input image

tion (NMT) or document summarization where the input and the output are a sequence of words, usually the input is encoded into a vector using a Recurrent Neural Network which is then fed to the decoder to generate the output sequence. Even though in theory architectures such as LSTM and GRU are designed to handle well dependencies in long sentences, in practice they still have problems when encoding long dependencies. Attention mechanisms come in to solve this issue. By having an attention mechanism in the encoder-decoder architecture, encoding the whole input as a single vector is no longer necessary, but rather at each time step the decoder input will be the weighted average of each word in the input sequence. This averaging is becoming part of the model to learn, in which the decoder will learn which parts of the input to attend to at each time step. An additional advantage of the attention mechanism is that it lets us interpret the model decisions by visualizing the attention as shown in Figure 2.4 .

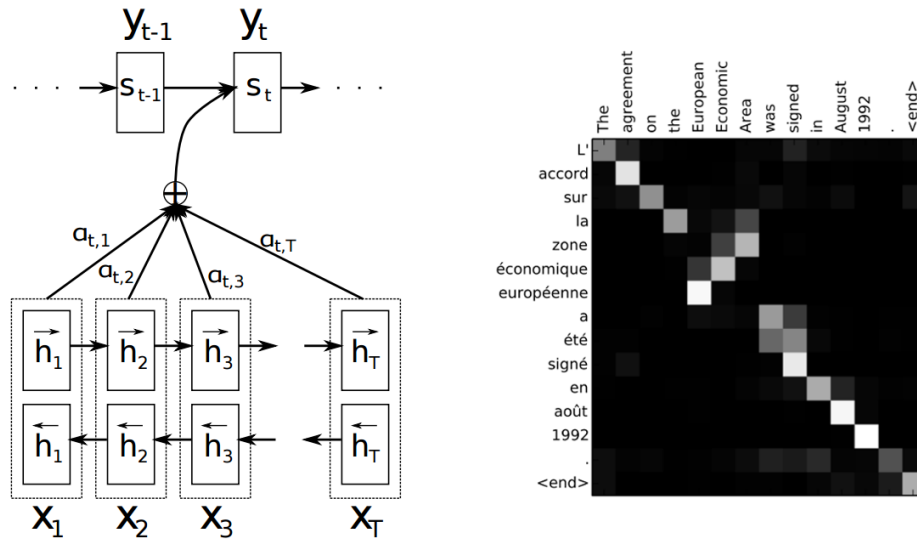


Figure 2.4: Figure from [5] showing implementation architecture of Attention mechanism for Seq2Seq models, next to it a visualization of output tokens attention weights to each input token for the task of Neural Machine Translation.

### 2.4.2 Training Sequence to Sequence models

After an input sequence is encoded, the decoding process starts with a "start of sequence" special token. The decoder generates a probability distribution over each token in the target vocabulary. The model is trained to maximize the likelihood of the target word at each time step given an input sequence for training.

The input to the decoder in the following time-step can be the predicted token by the decoder in the previous time step (i.e. the token with the highest probability in the output distribution), however this approach can lead to slow convergence and model instability. Thus the most common approach is to supply the observed sequence to the decoder at each time step, this approach is called teacher forcing [182]. Teacher forcing approach has proved to yield faster convergence during training sequence prediction models. However it leads to models being fragile when used in practice. Therefore there has been several adaptations to the teacher forcing algorithm one of them is the "Scheduled sampling" or "curriculum learning" approach to gently bridge the gap between

training and inference. The training starts by a teacher forcing approach and over time after several epochs and then the probability of the forced input is reduced to gradually teach the model how to deal with its own mistakes [10].

### 2.4.3 Sequence to Sequence models at inference time

At each time step during inference time, the seq2seq model is capable of generating probability distribution over the tokens of the target vocabulary, given an input sequence and the previously generated tokens  $P(Y_t|X, Y_1, \dots, Y_{t-1})$ . There are two ways to generate sequence from these conditional probabilities, either by random sampling or by trying to select the most likely sequence:

**Ancestral Sampling:** which means randomly sampling a sequence token by token according to the probability distribution.

$$\begin{aligned} & \text{While}(Y_{t-1} \neq \langle EOS \rangle) : \\ & Y_t \sim P(Y_t|X, Y_1, \dots, Y_{t-1}) \end{aligned} \tag{2.1}$$

**Greedy Inference:** this is usually done by feeding the token with the highest probability as an input to the next time step.

$$\begin{aligned} & \text{While}(Y_{t-1} \neq \langle EOS \rangle) : \\ & Y_t = \text{Argmax}(P(Y_t|X, Y_1, \dots, Y_{t-1})) \end{aligned} \tag{2.2}$$

It is important to note that Greedy inference is an approximation of the search space, meaning that it does not guarantee the highest probability sequence. This in practice will yield to the network preferring more probable words than rare more correct ones.

**Beam Search Decoding:** Beam search [89] for decoding Sequence to Sequence models [60, 164] is a generalization from the Greedy inference. A fixed number  $K$  of candidate sequences (beam) are kept those corresponds to the top  $K$  conditional probability. Beam search consists of two main sub processes at each time step: First, Beam expansion in which each incomplete candidate summary is fed to the decoder to generate the conditional probability at the next time step. Secondly Beam Selection where the conditional probability at each time step are augmented to their corresponding beam overall probability and only the top  $K$  beams are selected to be fed to the decoder separately at

the next time step. This process continues until all top beams terminate by an end of sequence special token  $\langle EOS \rangle$  become the defacto decoding algorithms for sequence prediction tasks due to its empirical gains over Greedy Decoding. Having said that, Beam search is  $K$  times slower than Greedy Decoding and requires keeping  $K$  copies of the decoder in the memory.

#### 2.4.4 Copy Actions

As shown in the previous sections, NLG models for tasks such as NMT and summarization usually generate sentences by emitting word by word at each time step. In which, at each time step the decoder output a probability distribution over words in the vocabulary. This probability distribution is usually calculated through a softmax function. Sometimes the target sentence to be generated is best expressed with words which do not appear as much or even never appeared in the training corpus, despite of having a sentence structure which is very easy to generate. For example, a basic seq2seq model for machine translation trained only on the European Parliament parallel corpora will probably struggle to translate sentences from social media posts, not only because of the domain shift but also because the decoder might need to output terms that is not usually discussed in the European Parliament and thus not contained in the output vocabulary. This problem is called the rare word problem and has been studied a lot [102, 62, 150, 92, 46]

One way to solve this problem is by compiling a vocabulary with a very large size, one that can possibly contain all words in the target language. However the nature of NLG systems cannot handle neither computationally nor in application very large vocabularies. If each time step the decoder has to select from a large number of words in the target vocabulary this will require time to calculate the denominator of the softmax which will affect the performance since the number of classes increases. Due to this computationally intensive nature of the softmax, It is a very common practice when building models for NLG to limit the input and the output vocabulary to the top most frequent words [102]. Words that are not from the top appearing words are replaced with a special tag  $\langle UNK \rangle$ . This makes any rare word become Unknown to the model as well.

Limiting the vocabulary for the top 30–80K words appearing in the training set might be suitable for the nature of machine translation or for specific corpora, where the top

30K words can cover adequate amount of information. When neural text generation started to expand to other areas, specifically for abstractive summarization and Question Generation, text rich with dates and numbers made the rare or the unseen word problem become more prevalent.

Having said that, these unknown words are not completely unknown, many of those words such as named entities, dates and numbers, can be seen in the input text that is given to be translated or summarized. Thus, many work has proposed to solve this problem using copy actions, in which unknown words can be copied directly from the input words instead of being generated from the output vocabulary.

en: The unk portico in unk ...  
 fr: Le unkpos<sub>1</sub> unkpos<sub>-1</sub> de unkpos<sub>1</sub> ...

Figure 2.5: An example from Luong et al. [102] showing one proposed way to model copy actions using special tokens from the input and output vocabulary.

Early work [102] modeled copy actions as a set of special tokens (as shown in Figure 2.5) to be added to the output vocabulary, after those special tokens are outputted their are being replaced with their corresponding words from the input text using 1-1 alignment between words in the input and output text. Later work by Gulcehre et al. [62] and See et al. [150] incorporated the copying mechanism in the seq2seq model itself by relying on pointer networks [173]. As shown in Figure 2.6, at each decoding time step the generation probability  $p_{gen}$  is being calculated which is the probability of generating a word from the output vocabulary. The inverse of this probability  $1 - p_{gen}$  is the copying probability. To decide which word from the input is going to be copied at each decoding time step, the pointer network outputs distribution over each position of the input words, those probabilities are those from the attention mechanism. Both generation and copying distributions are being weighted and summed to calculate the final distribution over the words to select from.

### 2.4.5 Evaluation of Natural Language Generation

Traditional evaluation methods for natural language generation systems [56] fall into two major classes: intrinsic or extrinsic [9]. Intrinsic evaluation seeks to evaluate directly



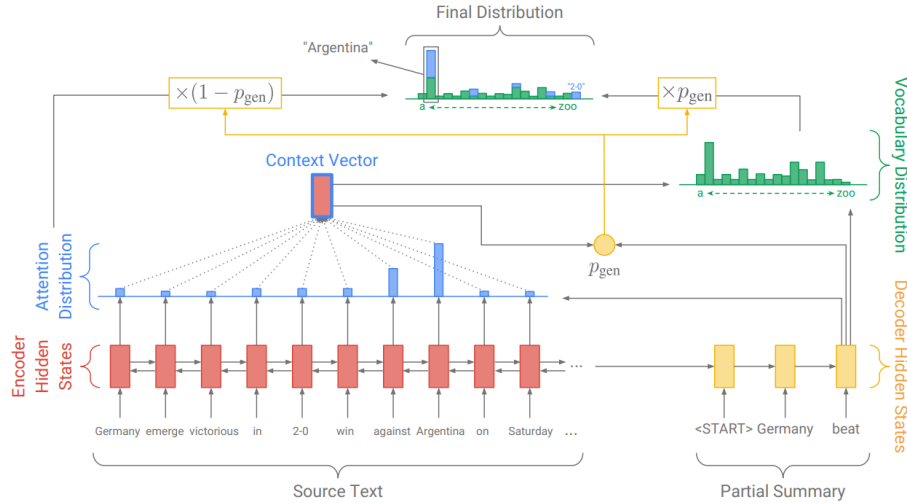


Figure 2.6: The pointer generator model from [150] showing the output probability distribution over output words is calculated using the regular generation mechanism with probability  $p_{gen}$  or through copying mechanism using the attention weights from the attention mechanism with probability  $1 - p_{gen}$ .

the generations done by the system to be evaluated, traditionally using human evaluators through directly evaluating aspects of the generated text such as fluency and correctness. While extrinsic evaluation on the other hand evaluates the effect of the NLG system when used for a certain application. During this part we will focus mainly on intrinsic evaluation.

### Automatic Evaluation

Automatic evaluation metrics have become the most common way in literature to evaluate NLG systems. According to Gkatzia et al.[56] up to 60% of the NLG literature in top NLP venues between 2012 and 2015 rely on automatic metrics. The most common forms of automatic metrics are word overlap metrics and semantic similarity metrics [128]. Both compare the generated text with equivalent corresponding text written by human experts. Below we show popular examples of both categories:

**BLEU** [130] is a metric that calculates the score of the generated sequence through measuring the number of n-grams of variable lengths that occur in the reference text. BLEU is one of the most if not the most used automatic metric for evaluating NLG systems

due to its reliability and simplicity. However, BLUE score is limited when it comes to measuring semantic or grammatical correctness of the output sentence.

**METEOR** [31] counts the number of exact matches between the NLG system and the reference similarly to the BLEU score. However, METEOR tries to match some of the mismatching tokens by using stemmers and lemmatizers with an addition of some penalties.

**ROUGE** [98] is a recall-oriented metric in which calculates the percentage of tokens in the system output that occurs in the reference sequence as well. ROUGE has several variations such as ROUGE-N which is calculated over the overlap of N-grams, ROUGE-L which is calculated over the Longest common subsequence (LCS), ROUGE-W is a weighted version of ROUGE-L that favors consecutive subsequences.

**CIDER** [172] is another metric developed originally for the image captioning task. CIDER is adapted to allow evaluation using multiple references, in which the weight of each n-gram overlap is adapted using a TF-IDF score to indicate the

### **Semantic Similarity Metrics:**

Even though word overlap automatic metrics like METEOR takes into consideration tokens that match in lemmas or the stem of the token, these metrics are oblivious to other forms of paraphrasing. Thus many work relied on additional metrics that calculate semantic similarity to evaluation NLG system rather than just doing a basic word overlap. Han et al. [67] devised a semantic similarity metric that is based on a distributional similarity using Latent Semantic Analysis augmented with semantic relations extracted from wordnet.

Embedding Greedy [144, 152] is another sentence similarity metric for evaluating NLG systems. The metric finds an alignment between tokens in the NLG system output and the reference sequence. This alignment is made to maximize the cosine similarity between pretrained embeddings of the aligned tokens. Afterwards, the sentence similarity score is computed as the mean cosine similarities between aligned tokens.

## 2.5 Tasks Related to Knowledge Graphs Relationships

In this section I'll list down several tasks related to knowledge bases. Most of these tasks are related to extraction or discovery of relation between world entities and representing them in a structured form in a knowledge base. Those tasks, though having lots of similarities, they slightly differ with regard to their inputs and hence affecting the kind of algorithms being used in such task. In the next subsections I'll discuss these differences in detail.

### 2.5.1 Open Relation (Information) Extraction

Information Extraction, Open information Extraction, Open Relation Extraction, OIE or OpenIE are all synonyms to the task which takes natural language sentences as input and extract semantic relations between entities in this sentence in a form of a triple  $\langle arg1; rel; arg2 \rangle$  [80].

Open IE has shown to be useful in many tasks such as document summarization [25], question answering [51] and knowledge base population [3]. A study by [160] has shown that relying on Open IE as an intermediate structure rather than other semantic structures, can enhance the performance of some tasks such as text comprehension, word analogy and word similarity.

The name *open* comes from the fact that OpenIE is not limited to a set of predefined relationships. This can be very powerful tool for exploring new domains in a way that is needless to have experts to compile a set of possible relation-types for each new domain. However, due to the nature of this task, outputs of OpenIE systems are rather shallow and can contain many semantic redundancies as well as uninformative and incoherent extractions. Arguments of OpenIE systems are represented with their surface forms from the input sentence. This means that out of all OpenIE extractions there can exist several attributes that refer to the same entity or relation type with different surface forms.

Considering the nature of this task where the target semantic relations are not specified before hand. A large body of literature have tackled this task in an unsupervised manner, some of these are fixed rule based of regular expressions over Part of Speech tags as in TEXTRUNNER [6] and REVERB [49]. In order to enhance recall of those rule based methods Mausam et al. introduced unsupervised pattern learning in OLLIE

```

OLLIE:
(1) (Republican candidate Mitt Romney; will be elected President in; 2008)[enabler=If he wins five key states]
(2) (Republican candidate Mitt Romney; will be elected; President)[enabler=If he wins five key states]
(3) (Mitt Romney; be candidate of; Republican)
(4) (Mitt Romney; be candidate for; Republican)
(5) (he; wins; five key states)

ReVerb:
(6) (he; wins; five key states)
(7) (Republican candidate Mitt Romney; will be elected President in; 2008)

PredPatt:
(8) (he; wins; five key states)
(9) (Republican candidate Mitt Romney; will be elected President in; 2008)

ClausIE:
(10) (he; wins; five key states)
(11) (Republican candidate Mitt Romney; will be elected; President in 2008 If he wins five key states)
(12) (Republican candidate Mitt Romney; will be elected; President in 2008)

OpenIE 5.0:
(13) (Republican candidate Mitt Romney; will be elected; President; T:in 2008)
(14) (he; wins; five key states)

Graphene:
(15) #1 CORE (Mitt Romney; will be elected; President)
("a) CONTEXT:NOUN_BASED Mitt Romney was a republican candidate .
("b) CONTEXT:TEMPORAL in 2008 .
("c) CONTEXT:CONDITION #3
("d) CONTEXT:NOUN_BASED #2
(16) #2 CORE (Mitt Romney; was; a republican candidate)
(17) #3 CONTEXT (he; wins; five key states)

```

Figure 2.7: Several OpenIE outputs for the same sentence "If he wins five key states, Republican candidate Mitt Romney will be elected President in 2008." from [125]

[107]. OpenIE systems them developed to make use some supervised algorithms based on noisy collected datasets such as Tree Kernels [191] or logic inference in STANFORD OPEN IE [3]. More recent details on the status of Open-IE systems and several variations are in this extensive survey by Niklaus et al. [125].

### 2.5.2 Relation Extraction (classification)

Relation extraction or relation classification is a very similar task to Open Information Extraction except that it operated over a predefined vocabulary of entities and Relationship types. The common input to this task is a sentence where two named entities are identified, while the output entity one of a predefined relations which represents the relation type being expressed in the input sentence between the two named entities being identified. This makes this task certainly more close to knowledge bases as the output of a Relation Extraction system can be directly injected into a knowledge base such as Wikidata or DBpedia. However at the same time it is very challenging to collect training dataset for such task to cover all possible relationships that exist in current open large scale knowledge bases. Contributions in this task have been in three directions:

**Firstly**, works that are concerned with automatic generation of training dataset for relation extraction. The Distant supervision assumption for Relation Extraction [112] has yielded a success in this direction by managing to create large datasets aligning knowledge base triples to sentences in free text. The assumption states that if two entities are connected through a semantic relationship in a knowledge base triple then every sentence including those two entities has high probability of expressing the same semantic relationship. As simple as it is [140] through a manual annotation experiment proved that it is correct 80% of the instances when applying this assumption on Wikipedia text with triples from Freebase. Several modification and variation of this assumptions has followed to either increase accuracy or recall on different text corpora [4, 47], One of the most commonly cited benchmarks for this setup is developed by Riedel et al. [139] which aligns sentences from the New York Times news corpus [148] with the Freebase [14] Knowledge base. Many models has been emerged from this benchmark dealing with the challenge of learning from noisy dataset either using feature engineering [112], graphical models [163] or neural networks with extra signals [171, 101].

**Secondly**, a slightly different benchmark [74] for the task has created another line of work, mainly referred to as relation classification. This task follows the same setup as relation extraction except that the training and evaluation datasets are of high quality and the set of target relationships are limited in size compared to the other setup by Mintz et al. [112]. This has led to another line of work that does not model noise in the training dataset but focuses more on how to learn better features through using deep neural networks either using Convolution Neural Networks [198, 35, 155, 177], Recurrent Neural Networks [200, 184] or Neural Networks with linguistic features [190, 187, 189].

**Thirdly**, as shown in chapter 1 emergence of new relation types is a serious matter to deal with. This can challenge models trained on classic setup of relation extraction from being able to detect new relation types this is because they rely on classifying a set of predefined relation types. This has led to the emergence of lines of work that can expand to new relation types that have not been defined during training time, early forms of this work relies on the usage of Universal Schemas [139] in which the set of relation types in a knowledge base is augmented by relations extracted from a large corpus of text. Recent work in this direction cast the relation extraction task in a Few shots [68] and a Zero shot setups [95].

### 2.5.3 Link Prediction or Knowledge Base Completion

By knowing enough information about a specific entity some other information can be easy to predict. For example knowing that a person is the president of united states it makes it very probable to predict where they were born or what is their nationality. Knowledge Base Completion, Relation Prediction, or Link prediction are all synonyms to the same task which is modeling that problem. This makes it another slightly different task than relation extraction, where the expression information in form of text is not necessarily given. The input to models trying to solve this task is set of triples in knowledge base and an incomplete triple  $\langle arg1, rel, ? \rangle$  and the model is required to replace the missing argument with the correct entity to complete this triple. An incredible amount of work has been published on this task by learning representations learning transitional representations of knowledge base entities and relations in vector space. In this body of work a knowledge base relationship are modeled as vector manipulations between entities in the vector space. Techniques to learn those representations varied from neural networks [156, 34] to Matrix factorization such as RESCAL [124] or in complex embeddings [167] or through transitional models such as TransE [16], TransH [178], TransD [185], DistMult [194]. Nickel et al. [123] survey provides an in depth review about learning representations for knowledge graphs.

### 2.5.4 Relation Discovery

The last task that operates over knowledge bases is Relation Discovery [195] or sometimes referred to as automatic ontology building. This task is concerned with helping building the structure of a knowledge base (ontology) from scratch, through extraction of new relation types from free text. Relatively little number of literature has tackled this task independently [199, 141] however it has been part of constructing very large knowledge base projects such as NELL [115] and Knowledge vault [34] without relying on a domain experts to identify and standardize the knowledge base ontology for each specific domain. Evaluation of relation discovery methods is very challenging especially when evaluating recall, this has led to the non-existence of standard methods of evaluating research techniques in this task.

In this chapter I introduced several tasks and line of works that represent the common background knowledge for tasks related to the interaction between knowledge bases and

Tasks	Benchmarks	Techniques + significant citations
<b>Information Extraction</b>	QASRL [159]	Self-supervision [6]
	Reverb45K [170]	Rule Based [49] Tree Kernels [191, 192] Logical inference [3]
<b>Relation Extraction</b>	NYT-FB [139]	Distant supervision [113, 4] Universal schemas [139] Neural Networks [171, 101, 163]
	FB15K-237 [1]	
	T-REx [47]	
	TAC KBP [77]	
	Google-RE <sup>1</sup>	
<b>Relation Classification</b>	SemEval-2010 Task 8 [74]	CNN [177, 155, 35]
		RNN [184, 200] RNN+linguistic features [181, 189, 187]
<b>Relation Prediction</b>	FB15K-237 [16] WN18RR [16]	Matrix Factorization [124]
		Transitional models [16, 178, 99]
		Text + KB embeddings [1]
		CNNs [32] Graphical models [133]
<b>Few-shot Relation Extraction</b>	FewREL [68]	Meta-Learning [68]
		GNN [68]
		Prototypical Networks [68]
<b>Zero-shot Relation Extraction</b>	Wikireading [75]	Reading comprehension [95]
<b>Relation Discovery</b>	–	Never ending learning [115]
		Bootstrapping from text [34]

Table 2.2: Table summarizing various tasks dealing with text and knowledge bases.

natural language. This chapter and the publications cited within can serve as a complementary reference to the in depth related work that will be introduced in each of the upcoming chapters. In the upcoming three chapters, I will present the main contributions of this thesis tackling the three main research questions introduced in chapter 1.



## Chapter 3

# Limitations of Knowledge Bases

### 3.1 Unsupervised Open Relation Extraction

As seen previously in Chapter 2, the task of relation extraction (RE) is the task of identification and classification of relations between named entities (such as persons, locations or organizations) in free text. RE is of utmost practical interest for various fields including event detection, knowledge base construction and question answering. Fig. 3.1 illustrates a typical RE task. For the first two sentences, RE should identify the semantic relation type *birth place* between the named entity pairs regardless of the surface pattern used to express the relation such as *hometown is* or *was born in*. RE should also distinguish it from the album production relation between the same named entities in the third sentence. Approaches towards RE varies between: (i) Supervised machine learning,

1. *David Bowie*'s hometown is *London*, United Kingdom.
2. *Axel Rose*, also known as "William Bruce", was born in *Lafayette*, Indiana.
3. *David Bowie* produced his first album in *London*, United Kingdom.

Figure 3.1: Sentences containing textual relations between *named entities*.

which requires large manually annotated datasets and typically suffers from the variety of surface forms for relations: although the first two sentences in Fig. 3.1 describe the birth city of a person, this is expressed in different words. (ii) Distant supervision [113]

employs named entities and relations mapped to the existing knowledge bases. As a result, the triple `<dbpedia.org/resource/David_Bowie, place_of_birth, dbo:birthPlace, dbpedia.org/resource/London>` for the aforementioned example would be extracted. However, distant supervision is limited to a fixed set of relations in the given knowledge base, which hinders adaptation to new domains.

Unsupervised approaches [196, 105] can potentially overcome these limitations of (distantly) supervised relation extraction by applying purely unsupervised methods enabling extraction of open relations (relations unknown in the knowledge base in advance). In this chapter, we propose an unsupervised approach to extract and cluster open relations between named entities from free text by re-weighting word embeddings and using the types of named entities as additional features.

### 3.1.1 Proposed Method

Our system builds sentence representations based on the types of the involved named entities, and the terms forming the relations. For the latter, we use pre-trained word embeddings after re-weighting them according to the dependency path between the named entities. These representations are clustered so that different representations of the semantically equivalent relations are mapped to the same cluster. As shown in Fig. 3.1, this approach would map the example sentences into two clusters, where the first one contains statements about birth places and the second one is focused on the album production.

Our evaluation shows that our system achieves a  $B^3$   $F_1$  score of 41.6% on the NYT-FB dataset [105], significantly outperforming the currently best performing state-of-the-art approaches based on variational autoencoders (by 16%). — Previous Version of Introduction, Start — Extracting triples from free text is a task of the utmost practical interest for Knowledge Base Construction and Completion [77, 115], and Question Answering [51]– to name a few. Fig. 4.1 presents an overview of our system for unsupervised open relation extraction, consisting of four stages: preprocessing, feature extraction, sparse feature reduction and relation clustering described in the following. **Preprocessing** For each sentence in the dataset, we extract named entities using DBpedia Spotlight [109] and consider all sentences containing at least two entities. For this set of sentences, the Stanford CoreNLP dependency parser is utilized to extract the lexicalized dependency path between each pair of named entities.

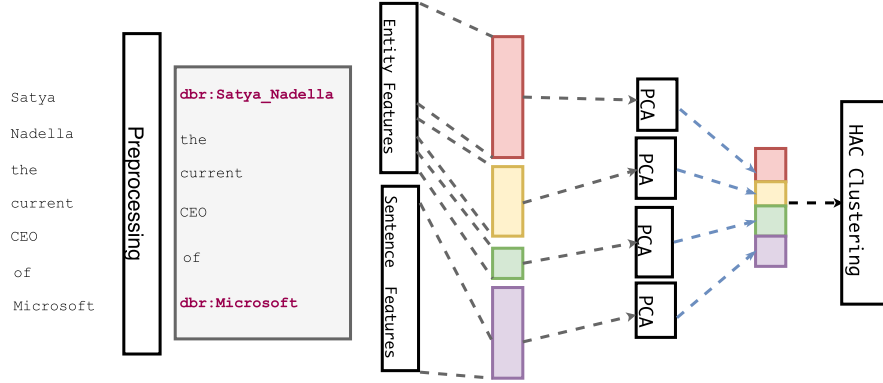


Figure 3.2: Our Proposed system for relation discovery and clustering

**Entity and Sentence Features Extraction** For each sentence, our method outputs a vector representation of the textual relation between each named entity pair. Features include word embeddings, dependency paths between named entities, and named entity types. Word embeddings provide an estimation of the semantic similarity between terms using vector proximity. Sentence representations are typically built by averaging word vectors. However, not all words in a sentence equally contribute to the expression of the relation between two named entities. Therefore we develop a novel method to re-weight the pre-trained word embeddings. Terms that appear within the lexicalized dependency path between the two named entities are given a higher weight. Intuitively, shorter dependency paths are more likely to represent true relationships between the named entities. The vector representation  $s(W, D)$  of each sentence is calculated through the following function:

$$s(W, D) = \sum_{w_i \in W} f(w_i, W, D) \cdot v(w_i), \quad f(w_i, W, D) = \begin{cases} \frac{C_{in} \cdot |W|}{|D|}, & \text{if } w_i \in D \\ C_{out}, & \text{otherwise} \end{cases},$$

where  $W = \{w_1, \dots, w_n\}$  is the set of terms in the sentence,  $D \subset W$  is the set of terms in the lexicalized dependency path between the named entities in the sentence, and  $v(w_i)$  is the pre-trained word embedding vector for  $w_i$ .  $C_{in} \geq 1$  and  $C_{out}$  are constant values experimentally set to 1.85 and 0.02. We use Glove trained on Wikipedia 2014 +

Gigaword 5 corpus <sup>1</sup> with word embeddings of size 100. As a baseline, we compare these representations with standard sentence representations features such as: TF-IDF, the sum of word embeddings, and the sum of IDF re-weighted word embeddings [138]. Intuitively, relations can connect entities of certain types. For example, a birth place relation connects a person and a location, although other relations between person and location are possible. Therefore, for each named entity, we use its DBpedia types and Stanford NER tags as features.

**Sparse Feature Reduction using PCA** Some of the features are more sparse than the others; concatenating them for each relation skews the clustering. In supervised relation extraction, this is not an issue as any learning algorithm is expected to do feature selection automatically using the training data. In unsupervised relation extraction there is no training data, hence we devise a novel strategy in order to circumvent the sparse features bias. Individual feature reduction of the sparse features is applied before merging them with the rest of the feature vectors. For feature reduction, we use Principal Component Analysis (PCA) [79].

**Relation Clustering using HAC** We use Hierarchical Agglomerative Clustering (HAC) to cluster the feature representations of each relation, with Ward’s [180] linkage criteria<sup>2</sup>, which yields slightly better results than the k-means clustering algorithm.

### 3.1.2 Evaluation

To evaluate our system, we use the NYT-FB dataset [105]. This dataset contains approximately 1.8M sentences divided into 80%-20% test-validation splits and aligned automatically to the statements (triples) from Freebase. The alignment between sentences and the properties of the Freebase triples in this dataset is considered as the gold standard for the relation clustering algorithm.

We use the validation split to tune the parameters for re-weighting word vectors and the PCA algorithm, and the test set for evaluating relation discovery methods. We compare our method using the best identified feature combination with the state-of-the-art models for unsupervised Relation Discovery, namely the variational autoencoders model

---

<sup>1</sup><http://nlp.stanford.edu/projects/glove/>

<sup>2</sup>accessing the clustering output by HAC at rank  $k$  giving  $k$  clusters

[105] and two other systems, Rel-LDA [196], and Hierarchical Agglomerative Clustering (HAC) baseline with standard features [197]. To make our results comparable we set the number of relations to induce (number of clusters  $k$ ) to 100, following the SOA systems.

Table 3.1 shows the performance of the clustering algorithm by relying only on sentence representations as features. Results demonstrate that our method of word embeddings re-weighted by the dependency path shows a significant improvement over other traditional sentence representations. Table 3.4 shows the performance when the dependency re-weighted word embeddings are merged with the rest of the proposed features and applying individual feature reduction. Our method outperforms the state-of-the-art relation discovery algorithm scoring a pairwise F1 score of 41.6%.

Feature	F <sub>1</sub>
TF-IDF	12.2
Word-Emb.	7.4
IDF-Emb.	10.3
Dependency Re-Weighted Emb.	<b>19.5</b>

Table 3.1: Comparison between different sentence representations when used as features for clustering.

Var. Autoencoder	Rel-LDA	HAC	Ours
35.8	29.6	28.3	<b>41.6</b>

Table 3.2: Pairwise F<sub>1</sub> (%) scores of different models on the test set of the NYT-FB dataset.

### 3.1.3 Conclusion

In the frame of this thesis we proposed a solution to tackle the first research question discussed in Chapter 1. To overcome the problem of rapidly evolving knowledge bases, we proposed an approach for unsupervised relation extraction from free text. Our method does not require any training examples and can generalize to unseen open relations. Our approach is based on a novel method of re-weighting word vectors according to the dependency parse tree of the sentence. As additional features, we use the types of named

entities involved in the relations. A final HAC clustering is applied to the sentence representations so that similar representation of a relation are mapped to the same cluster. Our evaluation results demonstrate that our method outperforms the state-of-the-art relation clustering method by 5.8% pairwise F1 score.

## 3.2 High Recall Open IE for Relation Discovery

The recent years have shown a large number of knowledge bases such as YAGO [162], Wikidata [176] and Freebase [14]. These knowledge bases contain information about world entities (e.g. countries, people...) using a set of predefined predicates (e.g. birth place, profession...) that comes from a fixed ontology. The number of predicates can vary according to the KB ontology. For example there are 61,047 DBpedia unique predicates compared to only 2,569 in Wikidata.<sup>3</sup> This has led to an emergence of unsupervised approaches for relation extraction which can scale to open relations that are not predefined in a KB ontology.

### Open Information Extraction

As explained previously in chapter 2, open information extraction (Open IE) systems extract linguistic relations in the form of tuples from text through a single data-driven pass over a large text corpus. Many Open IE systems have been proposed in the literature, some of them are based on patterns over shallow syntactic representations such as TEXTRUNNER [6] and REVERB [49], pattern learning in OLLIE [107], Tree Kernels [191] or logic inference in STANFORD OPEN IE [3].

Open IE has demonstrated an ability to scale to a non-predefined set of target predicates over a large corpus. However extracting new predicates (relation types) using Open IE systems and merging to existing knowledge bases is not a straightforward process, as the output of Open IE systems contains redundant facts with different lexical forms e.g. (*David Bowie, was born in, London*) and (*David bowie, place of birth, London*).

---

<sup>3</sup>as of April 2017

### Relation Discovery and Clustering

Relation clustering and relation discovery techniques try to alleviate this problem by grouping surface forms between each pair of entities in a large corpus of text. A large body of work has been done in that direction, through: clustering of OpenIE extractions [116, 119, 120], topic modeling [196, 197], matrix factorization [166] and variational autoencoders [105].

These approaches are successful to group and identify relation types from a large text corpus for the aim of later on adding them as knowledge base predicates.

### Relation Discovery with a Single Entity Mention

Previously described relation discovery techniques identify relations between a detected pair of named entities. They usually use a pre-processing step to select only sentences with the mention of a pair of named entities (Figure 3.3 example 1). This step skips many sentences in which only one entity is detected. These sentences potentially contain important predicates that can be extracted and added to a KB ontology.

Figure 3.3 illustrates different examples of these sentences, such as: When the object is not mentioned (example 2), Questions where the object is not mentioned (example 3) or when one of the entities is hard to detect because of coreferencing or errors in NER tagging (example 4). By analysing **630K documents** from the NYT corpus [148] as

1. The **official currency** of the **U.K.** is the **Pound sterling**.
2. The **U.K. official currency** is down 16 percent since June 23.
3. What is the **official currency** of **U.K.** ?
4. .. which is considered the **official currency** of **U.K.**

Figure 3.3: Examples of textual representations mentioning the predicate "official currency".

illustrated in Figure 3.4, the number of sentences with two 2 detected named entities is only **1.8M sentences**. Meanwhile, there are almost **30M sentences** with one entity (16 times more), which can be explored for predicate mentions. As the set of two-detected entities sentences is limited, so is the number of possibly discovered predicates.

We propose a predicate-centric method to extract relation types from such sentences while relying on only one entity mention. For relation clustering, we leverage various

features from relations, including linguistic and semantic features, and pre-trained word embeddings. We explore various ways of re-weighting and fusing these features for enhancing the clustering results. Our predicate-centric method achieves 28% enhancement in recall over the top Open IE system and with a very comparable precision scores over an OpenIE benchmark [159]. It demonstrates its superiority for the discovery of relation types.

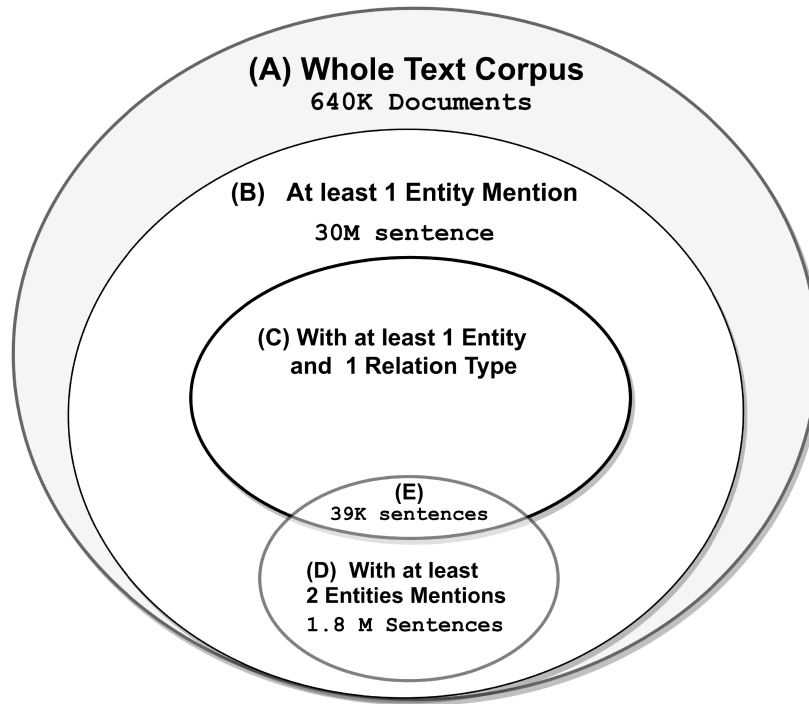


Figure 3.4: Distribution of sentences in the NYT corpus (A), which have: (B) at least 1 entity mention, (C) at least 1 entity and a predicate attached to it, (D) at least 2 entities mentions, (E) at least 2 entities and a relation in between in Freebase.

### 3.2.1 Our Approach

#### Extraction of Predicates

Banko et.al [7] show that the majority of relations in free text can be represented using a certain type of Part of Speech (POS) patterns (e.g. "VB", "VB IN", "NN IN"). Additionally Riedel et al. [139] propose the Universal Schemas model in which the lexicalized dependency path between two named entities in the same sentence is used



to represented the relation in between. We follow a similar approach to extract lexical forms of predicates in sentences and connect them to named entities in the sentences.

First to expand the set of predicate patterns proposed by Banko et al., we collect labels and aliases for 2,405 Wikidata [176] predicates, align them with sentences from Wikipedia, and run the CoreNLP POS tagger [103] on them. This results in a set of 212 unique patterns  $POS = \{pos_i, \dots, pos_n\}$ <sup>4</sup>.

Second, for each sentence in the corpus we do the following:

- (i) extract the linguistic surface forms of predicate candidates  $P_c$  by matching the POS tagging of the sentence with the set of POS patterns  $POS$ .
- (ii) extract candidate named entities  $E_c$  using the CoreNLP NER tagger [103].
- (iii) extract the lexicalized dependency path  $dp_i$  and its direction between every named entity  $e_i \in E_c$  and candidate relation predicates  $p_i \in P_c$  (if exist). The direction of the dependency path highly correlates with the entity being subject or object of the candidate predicate [142].

The result of this process is a set of extractions  $Ext = \{(p_i, e_i, dp_i) \dots (p_n, e_n, dp_n)\}$ , in which a predicate  $p_i$  is connected to a named entity  $e_i$  through a directed dependency paths  $dp_i$ . We ignore all the candidate predicates that are not connected to a named entity though a dependency path. The confidence for each extraction is calculated according to the rank of its dependency path  $dp_i$  and its POS pattern.

### Predicates Representation and Clustering

For each predicate in  $Ext$ , there are predicates though having different surface forms, express the same semantic relationship (e.g. "was born in", "birth place"). Following [116], we treat predicates with the same surface form as one input to the clustering approach. A feature representation vector for each unique predicate is built from multiple sentences across the text corpus. In the literature, this approach is referred to as the macro scenario, in contrast to the micro scenario [196, 105] where every sentence in the corpus is treated individually. The input to the clustering process in the macro scenario is very small in comparison to the micro scenario, which makes the macro scenario more

---

<sup>4</sup><http://bit.ly/2obhbyF>

scalable.

For each unique predicate  $p_i \in P$  we built a feature vector that consists of the following set of features:

1. Sum of TF-IDF re-weighted word embeddings for each word in  $p_i$ .
2. Count vector of each entity appearing as subject and as an object to  $p_i$
3. Count vector of entity types appearing as subject and as an object to  $p_i$
4. Count vector of each unique dependency path  $p_i$  that extracted  $p_i$
5. The POS pattern of  $p_i$  encoded as a vector containing counts of each POS tag.

The previous features are not equally dense – concatenating all of them as a single feature vector for each relation is expected to skew the clustering algorithm. In supervised relation extraction, this is not an issue as the learning algorithm is expected to do feature selection automatically using training data. Here, it is not the case. In order to circumvent the sparse features bias, we apply individual feature reduction of the sparse features before merging them to the rest of the feature vectors. For feature reduction, we use Principal Component Analysis (PCA) [79]. Once this reduction is applied, we apply a K-Means clustering [69] algorithm over the relations feature vectors in order to group relations into  $k$  clusters.

Sentence	Target predicate	Predicate Centric Extraction
Nicephorus Xiphias , who had conquered the old Bulgarian capitals.	conquered	conquered → dobj → MISC
Muncy Creek then turns northeast , crossing Pennsylvania Route 405	crossing	crossing → dobj → LOCATION
This was replaced by a Town Hall	replaced by	replaced by → nmod → LOCATION
Starting in 2009 , Akita began experiencing ...	Starting in	Starting in → nmod → DATE

Table 3.3: Example sentences where all OpenIE systems failed to extract target relations, and their corresponding Predicate-Centric extractions.

### 3.2.2 Experiments and Evaluation

#### Predicates Extraction

We demonstrate the effectiveness of using the proposed predicate-centric approach for relation discovery. For that we use a large scale dataset that was used for benchmarking Open IE [159]. The dataset is comprised of 10,359 Open IE gold standard extractions

over 3,200 sentences. Extractions are evaluated against the gold standard using a matching function between the extracted predicate and candidate predicates from Open IE systems. Extracted predicates that do not exist in the gold standard are calculated as false positives. We compare our predicate extraction method with a set of 6 Open IE systems, which are: REVERB, OLLIE, STANFORD-OPENIE, CLAUSIE [28], OPENIE4.0 an extension of SRL-based IE [24] and noun phrase processing [129], and PROPS [161]. Figure 3.5 shows that our proposed approach scores the highest recall amongst all the Open IE systems with 89% of predicates being extracted, achieving 28% improvement over CLAUSIE, the Open IE system with the highest recall and 6% improvement in precision over the same system. This shows that our approach is more useful when the target application is relation discovery, as it is able to extract predicates in the long tail with comparable precision, as shown in Figure 3.6. Table 3.4 shows a set of example sentences in the evaluation dataset in which none of the existing Open Information Extraction systems were able to extract, while they are correctly extracted by our approach.

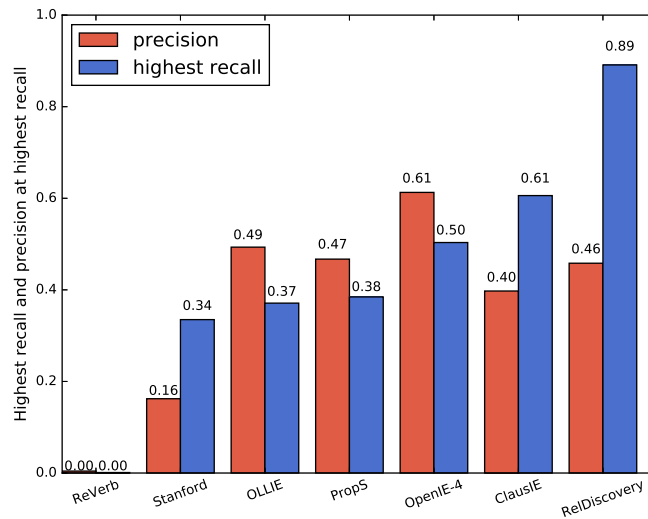


Figure 3.5: Maximum recall of top Open IE systems and their corresponding precisions in comparison with our approach **RelDiscovery** on [159] evaluation dataset.

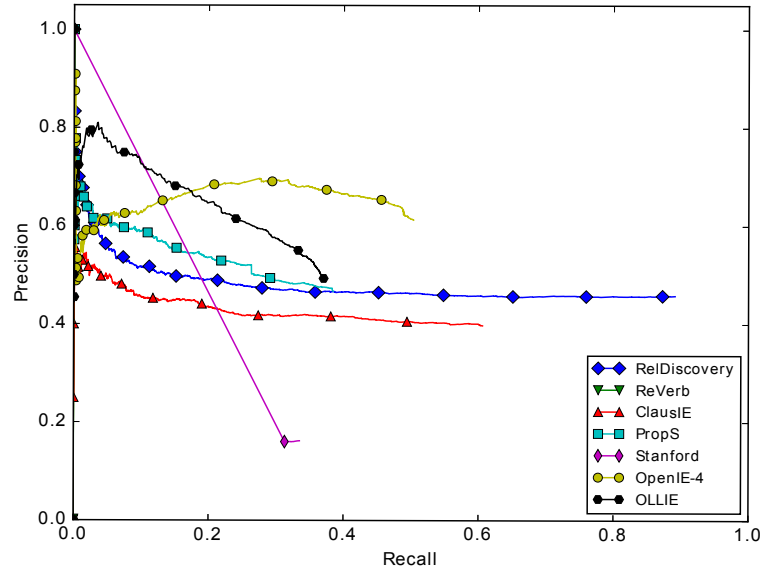


Figure 3.6: Precision and recall curve of our relation discovery method **ReIDiscovery** with different OpenIE systems.

### Quality of Relation Clustering

To the best of our knowledge, the literature does not provide datasets for evaluating Relation Discovery methods on the macro scenario. So we use *GOOGLE-RE*<sup>5</sup>, a high quality dataset, that consists of sentences manually annotated with triples from Freebase [14]. The dataset consists of 34,741 labeled sentences, for 5 Freebase relations: "institution", "place of birth", "place of death", "date of birth" and "education degree". We run our predicate extraction approach on the dataset and manually label the most frequent 2K extracted relations into 6 classes: the 5 target semantic relations in *GOOGLE-RE* and an additional class "OTHER" for other relations. We then divide them to 80-20% test-validation splits. For feature building, we use word2vec pre-trained word embeddings [110]. We tune the PCA using the validation dataset. Results in Table 3.4 show that the re-weighting of Word embedding using TF-IDF had a significant improvement over only summing word embeddings. This opens the door for exploring more common unsupervised representations for short texts. Additionally, individual feature reduction on the sparse features has significantly enhanced the pairwise F1 score of the clustering algorithm.

<sup>5</sup><http://bit.ly/2oyGBcZ>

Em-Ft	wEm-Ft	wEm-Ft-PCA	ALL
0.41	0.50	0.55	<b>0.58</b>

Table 3.4: pairwise F1 scores using word embeddings and sparse features (Em-Ft), after re-weighting word embeddings (wEm-Ft), after doing feature reduction (wEm-Ft-PCA), and combining all features (ALL).

### 3.2.3 Conclusion

In this subsection, we introduced a high recall approach for predicate extraction with the potential to cover up to 16 times more sentences in a large corpus. Our approach is predicate-centric and learns surface patterns to directly extract lexical forms representing predicates and attach them to named entities. Evaluation on an OpenIE benchmark shows that our system was able to achieve a significantly high recall (89%) with 28% improvement over the CLAUSIE, the Open IE system with the highest recall. It shows also a with very comparable precision with the rest of the OpenIE systems. Additionally, we introduce a baseline for comparing similar predicates. We show that re-weighting word embeddings and performing PCA for sparse features before fusing them significantly enhances the clustering performance, reaching up to 0.58 pairwise F1 score.

One of the main problems discussed in our 2nd research question in chapter 1 is the lack of datasets aligning knowledge bases and natural language. This problem was also manifested in in this chapter during the evaluation of our techniques. Although our proposed contributions managed to surpass the state-of-the-art in their respective tasks, many of the standard benchmarks available were of limited size and coverage, this posed many challenges. In this regard, in the next chapter I will introduce several contributions to help fixing the problem of dataset limitations.

## Chapter 4

# Limitations of Training Data

### 4.1 T-REx: A Large Scale Alignment of Natural Language with Knowledge Base Triples

Reducing the gap between Natural Language and structured Knowledge Bases (KB) has been the concern of many research tasks such as: Relation Extraction, KB Population, Natural Language Generation from KB triples and Question Answering. Models built for these tasks rely on training datasets containing alignments between sentences in free text and KB triples. The efficiency of such models and their ability to generalize, rely heavily on the quality, size and coverage of the datasets being used for their training and validation. Previous works [113, 196] have created free text / KB entries alignments either manually or automatically for the purpose of training and evaluation of their models. Still, available alignments suffer from several shortcomings [4, 106] : 1) limited size in terms of the number of alignments, 2) limited coverage where the number of represented predicates is not enough to generalize to larger domains, and/or 3) low or unreported quality. s4) reusability issues where many of the alignments are created for a specific task and are not published in a format that is suitable for other tasks. Several works in the literature have pointed out these shortcomings and have shown the importance of building a high-quality large scale alignments [4, 106].

In this work, we present *T-REx* a large scale dataset that contains large scale alignments

between free text documents and KB triples. *T-REx* is made up of 3.09 million Wikipedia abstracts aligned to 11 million Wikidata triples, covering more than 600 unique Wikidata predicates. This makes it two orders of magnitude larger than the largest available alignments to the community and covers 2.5 times more predicates. *T-REx* is built through leveraging techniques from distant Supervision, Relation Extraction and information retrieval. In order to alleviate reproducibility problems from previous methods for creating alignments, we define the customizable architecture of the alignment pipeline. This pipeline uses three different automatic alignment techniques aiming at increasing the alignments coverage. We evaluate the quality of *T-REx* by running a crowd-sourcing experiment over 2,600 created alignments. The best automatic alignment technique in *T-REx* achieved an accuracy of 97.8% over the evaluated subset of the dataset. *T-REx* is publicly available at <https://w3id.org/t-rex>.

### 4.1.1 Related Work

A considerable body of work has created alignments between free text and KB triples. In this section we take a look on the most popular datasets; we compare their size and coverage. The *TAC-KBP* dataset is built from news wire and web forums. The dataset is generated as a bi-product of the evaluation process of the TAC KB population competition<sup>1</sup>, where human annotators evaluate the output of each competing system. The dataset is limited in size as it consists in 5 classes and 41 predicates. Moreover its quality and coverage depend on the quality of the competing systems. A larger body of work has targeted automatic building of alignments for relation extraction through distant supervision. Several work [113, 196] have aligned the New York Times corpus with Freebase triples, resulting in as many variations of the same dataset, *NYT-FB*. This dataset is prone to bias and coverage issue since the Named Entity linking used for its construction is based on keyword matching against Freebase labels. For example, the NYT-FB version built in [196] contains almost 39K alignments for 258 Freebase properties. 30.7% of those alignments are for the sole predicate `freebase:location/country`. Additionally the New York Times corpus is not fully publicly available, which hinders the replication of research work based on this dataset.

---

<sup>1</sup><http://bit.ly/tackbpcpetition>

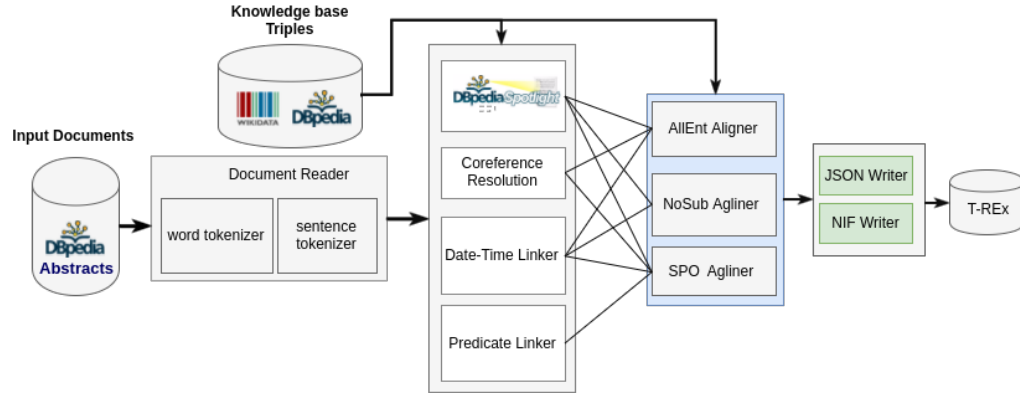


Figure 4.1: Overview of the alignment pipeline and its components

The *FB15K-237*<sup>2</sup> dataset [1] contains alignments of the Clueweb dataset with Freebase named entities [53] and Freebase triples. The dataset is of relatively large size (2.7 million alignments) though it lacks the original text from which the alignments are derived – This makes it unsuitable for some applications such as natural language generation. *Google-RE*<sup>3</sup> is a Google dataset with 60K sentences from Wikipedia, manually aligned with Freebase. Despite its high-quality, the dataset is labeled for only five Freebase relations. *WikiReadings* [75] is another dataset containing rough alignments created by replacing each subject of a Wikidata triple by the whole text of its Wikipedia article. Despite its large size, the dataset does not contain actual alignments between text and KB triples as there is no way to tell whether all the mentioned triples appear in the text, nor, if applicable, their location in the original text. Table 4.1 lists different alignments with their size and coverage.

Dataset	Documents / Format	Unique predicates	Aligned Triples	Available
NYT-FB	1.8M sent.	258	39K	partially
TAC KBP	90K sent.	41	122K	closed
Google-RE	60K sent.	5	60K	publicly
FB15K-237*	2.7 M patterns	237	2.7M	publicly
Wikireadings	4.7M articles	884	n.a.	publicly

Table 4.1: : Statistics over existing alignments from previous work.

<sup>2</sup><https://www.microsoft.com/en-us/download/details.aspx?id=52312>

<sup>3</sup><https://code.google.com/archive/p/relation-extraction-corpus>



### 4.1.2 *T-REx* Creation

#### Alignment pipeline

T-REx creation pipeline (Figure 4.1) contains components for document reading, entity extraction, triple alignment and dataset exportation into different formats. We paid attention to make our pipeline customizable and allow other work to insert their own components or to use it over other text corpora.

**Document Reader:** It gets documents from a dump and outputs in an format readable by all components. Also, it includes sentence and word tokenizers.

**Entity Extraction:** For each document, we use extracted name entities in the text and link them to their URI with the DBpedia Spotlight [109] entity linker.

**Date and Time Extraction:** We use the Stanford temporal tagger SUTime [19] to extract temporal expressions and their locations in documents. We normalize them to the XSD Date and Time Data Type format as expressed in most KB.

**Predicate Linking:** A sentence is more likely to express a KB triple if the label of the predicate forming this triple matches with any sequence of words in that sentence. A predicate linker links a sequence of words in a paragraph to its equivalent KB predicate URI if it matches the predicate label or any of its aliases in the KB.

**Coreference Resolution:** We use the Stanford CoreNLP co-reference resolution component [104]. Additionally we provide a robust heuristic inspired from [4]. We map a list of possible pronouns to each KB entity according to values of specific predicates such as "gender" and "instance of". Then, we link each pronoun in a sentence to its document main entity if they map.

**Triple aligners:** Triple aligners are the main components of our pipeline: each provided document is aligned with a set of KB triples expressed in the document alongside with their locations. They are described in the next subsection.

**Document Writers:** They export documents with annotation in standard formats. We propose a plain JSON format and NIF 2.0 [73], a RDF/OWL-based standard annotation format for natural language processing.

### Triple aligners

Let  $t_{xyz} = (e_x, e_y, e_z) \subset \mathcal{EXPXE}$  be one of all possible triples in a KB where  $\mathcal{E} = \{e_i, \dots, e_n\}$  and  $\mathcal{P} = \{p_i, \dots, p_n\}$  be the sets of all entities and properties represented in the KB respectively. Given a corpus of text documents, each document  $d$  contains a set of sentences  $d = \{s_i, \dots, s_n\}$ , a main entity  $e_{doc}$  and a set of linked entities  $\mathcal{E}_{doc} = \{\mathcal{E}_i, \dots, \mathcal{E}_n\}$  where  $\mathcal{E}_i$  is the set of entities linked in sentence  $s_i$ .

Following [4] we explore different methodologies to create those alignments using the distant supervision assumption. Distant supervision creates a set of alignments  $A$  between all triples whose subject and object entities are in the set of tagged entities in this sentence. i.e.  $A = \{(s_i, t_{xyz}) \mid e_x \in \mathcal{E}_i \wedge e_z \in \mathcal{E}_i\}$ .

**NoSub aligner:** In practice the subject entity is usually mentioned once at the beginning of the paragraph and is often referred implicitly or using pronouns. These implicit lexicalizations can hardly be detected by entity linkers, and lead to a coverage issue. The NoSub aligner relaxes the distant supervision assumption and assumes that sentences in one paragraph often have the same subject. It extracts a set of alignments  $A = \{(s_i, t_{xyz}) \mid (e_x = e_{doc} \wedge e_z \in \mathcal{E}_i) \vee (e_z = e_{doc} \wedge e_x \in \mathcal{E}_i)\}$ . This relaxation comes at a price: the position of the subject entity in each aligned triple is not known as the aligner assumes it is implicitly mentioned. And finally it assumes that all aligned triples have the paragraph main entity as their subject or object. This is not always the case, e.g. Table 4.1.2 Example 5 where "Brixton, London" can be mapped to the triple  $(dbr:Brixton, dbr:London)$  even if both entities are not the main topic of the paragraph.

**AllEnt aligner:** another annotation methodology in which every pair of entities in a sentence is considered in alignment and mapped to their equivalent KB relations. For implicit mentions of entities, we use co-reference resolution to extract all mentions of the main entity of the paragraph. Given  $\mathcal{E}' = \mathcal{E}_i \cup \mathcal{E}_i^{coref}$  the union of the sets of entities in the sentence through named entity linking and co-reference resolution, AllEnt extracts a set of alignments  $A = \{(s_i, t_{xyz}) \mid e_x \in \mathcal{E}' \wedge e_z \in \mathcal{E}'\}$ .

**SPO aligner:** The alignment of every pair of entities as shown in Table 4.1.2 Examples 8 & 9 can sometimes be noisy: it aligns triples that are not necessarily mentioned in the sentence. For that, the SPO aligner aligns triples not only when the subject and object of a triple are mentioned in a sentence but also when the predicate of the triples has been extracted. Given  $\mathcal{P}_i \subset \mathcal{P}$  the set of predicates tagged in

the sentence  $s_i$  using the predicate linker, the SPO aligner creates a set of alignments

$$A = \{(s_i, t_{xyz}) | e_x \in \mathcal{E}_i \wedge p_y \in \mathcal{P}_i \wedge e_z \in \mathcal{E}_i\}.$$

# Triples	NoSub	AllEnt	SPO
1) wd:David_Bowie wdt:nationality wd:England .	x	x	
2) wd:David_Bowie wdt:occupation wd:singer .	x	x	
3) wd:David_Bowie wdt:occupation wd:Actor .	x	x	x
4) wd:David_Bowie wdt:birthPlace wd:Brixton .	x	x	
5) wd:Brixton wdt:region wd:London .		x	
6) wd:David_Bowie is wdt:child_of wd:Margaret_Mary .	x	x	x
7) wd:David_Bowie is wdt:child_of wd:Haywood_Stenson .	x	x	x
8) <i>wd:Margaret_Mary wdt:Divorce wd:Haywood_Stenson .</i>		x	
9) <i>wd:Margaret_Mary wdt:deathPlace wd:London .</i>		x	

Table 4.2: Comparison between different extractions of three alignment schemes for a sample paragraph of two sentences. The detected properties in the paragraph are put between square brackets. Wrong alignments are in italic.

### 4.1.3 T-REx Dataset

The *T-REx* dataset is obtained by running the alignment pipeline to create large scale alignments between the Wikipedia Abstracts Dataset and Wikidata triples. We feed the pipeline with documents from the Wikipedia Abstracts dataset [18], an open corpus of annotated Wikipedia texts. We use its English section, containing 4.6M text documents. As a source of triples, we use the Wikidata truthy dump<sup>4</sup> containing 144M triples. Wikidata is an open collaborative KB, created and maintained by a large number of volunteers. The result of the alignment process is *T-REx*, a large dataset with alignments of KB with free text, provided from the three alignment techniques previously presented.

#### Size and Coverage

In Table 4.3 we compare the number of alignments in the *T-REx* dataset with the largest datasets of the literature NYT-FB and TAC-KBP. All of the 3 alignment techniques proposed in *T-REx* have reported a substantial larger number of alignments than the two

<sup>4</sup><https://dumps.wikimedia.org/wikidatawiki/entities/20170503/>

	# Documents	Alignments	Numerical Alignments	Uniq predicates
NYT-FB	1.8M	39K	None	258
TAC-KBP	0.09M	122K	n.a.	41
<i>T-REx</i> _SPO	0.79M	1.2M	21K	336
<i>T-REx</i> _NoSub	2.85M	5.2M	561K	642
<i>T-REx</i> _AllEnt	<b>3.09M</b>	<b>11.1M</b>	<b>350K</b>	<b>633</b>

Table 4.3: Number of alignments in different datasets

other datasets. The largest number of alignments was achieved by the AllEnt aligner with 11.1M alignments. In terms of coverage, the NoSub aligner recorded 642 predicates. This makes *T-REx* two orders of magnitude larger than the largest available alignments, representing 2.5 times more predicates. Moreover, having a significant number of examples for each predicate is of the utmost practical interest for training high coverage models, regardless the NLP task at hand. In Figure 4.2, we illustrate the gap between *T-REx* and prior datasets on the predicate coverage criteria by plotting the distribution of the number of alignments created for each predicate. *T-REx* has substantially more examples than the other datasets, not only for the most common predicates but also for the long tail ones.

### Availability and Licensing

*T-REx* and its alignment pipeline are publicly available<sup>5</sup> under a Creative Commons Attribution-ShareAlike 4.0 International License. *T-REx* is available on the following persistent address <https://w3id.org/t-rex> and registered at Datahub <https://datahub.io/dataset/t-rex>. *T-REx* is available to download in two formats: JSON and RDF following NIF 2.0 format. Each alignment in *T-REx* is described and enriched with additional metadata to guarantee *T-REx* reusability and suitability for different tasks. Annotations and how to use the dataset are described in detail in the *T-REx* webpage.

#### 4.1.4 Evaluation

In order to evaluate the quality of *T-REx* we have led a crowdsourcing experiment on a subset of the alignments comprised of 2,600 aligned triples distributed over our three

<sup>5</sup>[http://bit.ly/trex\\_alignments](http://bit.ly/trex_alignments)

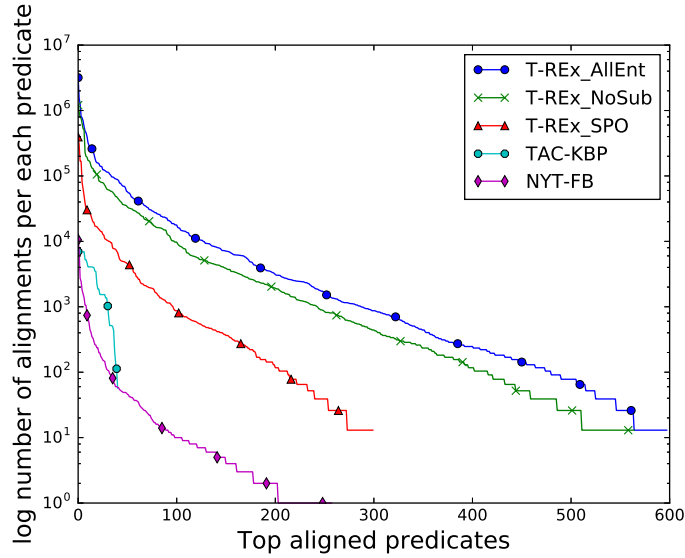


Figure 4.2: Distribution of the number of alignments created for each predicate

alignment techniques from 700 Wikipedia abstracts. We asked contributors<sup>6</sup> to read each document carefully and annotate each alignment to be true only if the triple is explicitly mentioned in the given document. Each alignment is being annotated at least 5 times. For example, given the sentence "*Jonathan Swift was born in Dublin, Ireland*", the triple "Ireland, Capital of, Dublin" should be annotated as False as it is not directly implied from the sentence despite the fact itself being correct. To guarantee high-quality annotations, we manually annotated 100 documents and used them to filter out spammers and non-qualified contributors. One of each 4 questions given to a contributor contains a test question, contributors who score less than 80% accuracy on these questions were disqualified from the crowdsourcing experiment.

Table 4.4 shows the accuracy of each alignment methodology and its corresponding inter annotator agreement  $I$ , calculated through the following formula:

$$I = 1 - \frac{\sum_{i=0}^N \left| \frac{f_i}{a_i} - t_i \right|}{N} \quad (4.1)$$

where  $t_i \in \{0, 1\}$  is the value of the majority vote for the alignment  $i$ ,  $f_i \in [0, a_i]$  is the number of times the alignment was labeled as True and  $a_i$  is the number of manual

<sup>6</sup>Instruction page: <http://bit.ly/2pBOZpx>

	T-REx_AllEnt	T-REx_SPO	T-REx_NoSub
Accuracy	0.88	0.957	0.978
Inter-Annotator	0.854	0.926	0.962

Table 4.4: Accuracy of each alignment methodology in T-REx

Property Label	T-REx_AllEnt	T-REx_SPO	T-REx_Nosub	Inter ann.
located in	0.949	1.0	1.0	0.9
member of sports team	1.0	0.997	0.99	0.97
date of birth	1.0	1.0	1.0	0.967
date of death	1.0	1.0	0.989	0.978
country of citizenship	0.91	1.0	0.95	0.923
educated at	0.875	0.92	1.0	0.916
occupation	0.9	0.94	1	0.93
spouse	0.75	0.94	1.0	0.916
capital	<b>0.4</b>	<b>1.0</b>	n.a.	0.82

Table 4.5: Accuracy of top properties for each annotation methodology in T-REx

annotators for it.  $N$  is the total number of alignments being annotated.

The NoSub aligner has scored the top accuracy scoring 97.8%, compared to 95.7% for the SPO aligner, let alone that the Nosub aligner has almost 4 times more extractions. Notice that the NoSub aligner relies on the assumption that the main entity of the paragraph is already known. This works efficiently on Wikipedia abstracts where the main subject is predetermined beforehand. However this might not be the case when using other target text corpora such as news documents. However, the SPO aligner has the advantage of extracting the positions of the subject, predicate and the object in the text, which makes it more suitable for training extractive models for Relation Extraction and Question Answering [75, 137]. [75, 137, 20]. Table 4.5 shows the alignment accuracy of top occurring predicates along side with inter annotator agreement.

#### 4.1.5 Error Analysis

In order to investigate more when the distant supervision assumption falls with respect to accuracy, we handpicked a sample of wrong alignments to analyze their main causes. We noticed three main causes of alignment errors: 1) Nested relations errors, where multiple relations in a short sentence share the same entities e.g. Table 4.6 example 1. This can be alleviated by creating aligners who take into consideration the linguistic

Alignment	Cause of error
1) He was the son of Ekoji I as well as the younger brother of Serfoji I. (Ekoji I, child(ren), Serfoji I)	Nested Relations
2) Ernst Gustav Kuhnert was born in Tallinn, Estonia (Tallinn, Capital of, Estonia)	Wrong Entailment
3) Carolyn Virginia Wood (born December 18, 1945) is an American.. (Virginia, country, American)	Entity Linking

Table 4.6: Causes of error in alignments

structure of the sentence such as dependency paths. 2) Wrong entailment, where the aligners aligns triples that do not imply the sentence, as shown on example 2 in Table 4.6. Here, the sentence describes the predicate *located in* but the aligned predicate was a sub property of it *capital of*. This can be alleviated through incorporating implication rules in the alignment process [30]. 3) Entity linking errors like in example 3 of Table 4.6. Alleviating these three main types of errors is the main future directions of T-REx enhancement.

#### 4.1.6 Conclusion

In this section we presented *T-REx* a large scale alignments between Wikipedia abstracts and Wikidata Triples. *T-REx* consists of 3 types of alignments made by 3 automatic alignment methodologies. Each of them provides a significantly larger number of alignments and covers higher number of predicates than any of the existing datasets. Through an extensive crowd-sourcing evaluation we managed to measure and assure that *T-REx* is of high quality, and perform an error analysis to understand when the common cases where distant supervision fails to produce high quality alignments. All efforts to produce *T-REx* and developing the framework behind were crucial for the continuation of this thesis, as any efforts on topics including knowledge bases and natural language interactions were challenged by the lack of training and evaluation high quality datasets. During the course of this thesis and as will be shown in the next chapters, *T-REx* and its underlying framework were used to provide datasets for training and evaluation of our models.

To continue tackling our 2nd research question about the lack of dynamic ways of generating training datasets for Question Answering systems. This is a slightly different

type of dataset which aligns knowledge base triples with Questions in Natural Language – unlike *T-REX* where KB triples are aligned with evidence statements in natural language.



## 4.2 Zero-Shot Question Generation from Knowledge Graphs for Unseen Predicates and Entity Types

Questions Generation (QG) from Knowledge Graphs is the task consisting in generating natural language questions given an input knowledge base (KB) triple [152]. QG from knowledge graphs has shown to improve the performance of existing factoid question answering (QA) systems either by dual training or by augmenting existing training datasets [33, 85]. Those methods rely on large-scale annotated datasets such as SimpleQuestions [15]. Building such datasets is a tedious task in practice, especially to obtain an unbiased dataset – i.e. a dataset that covers equally a large amount of triples in the KB. In practice many of the predicates and entity types in KBs are not covered by those annotated datasets. For example 75.6% of Freebase predicates are not covered by the SimpleQuestions dataset <sup>7</sup>. Among those we can find important missing predicates such as: `fb:food/beer/country`, `fb:location/country/national_anthem`, `fb:astronomy/star_system/stars`.

One challenge for QG from knowledge graphs is to adapt to predicates and entity types that were *not* seen at training time (Zero-Shot Question Generation). Since state-of-the-art systems in factoid QA rely on the tremendous efforts made to create SimpleQuestions, these systems can only process questions on the subset of 24.4% of freebase predicates defined in SimpleQuestions. Previous works for factoid QG [152] claims to solve the issue of small size QA datasets. However encountering an unseen predicate / entity type will generate questions made out of random text generation for those out-of-vocabulary predicates a QG system had never seen. We go beyond this state-of-the-art by providing an original and non-trivial solution for creating a much broader set of questions for unseen predicates and entity types. Ultimately, generating questions to predicates and entity types unseen at training time will allow QA systems to cover predicates and entity types that would not have been used for QA otherwise.

Intuitively, a human who is given the task to write a question on a fact offered by a KB, would read natural language sentences where the entity or the predicate of the fact occur, and build up questions that are aligned with what he reads from both a lexical and grammatical standpoint. We propose a model for Zero-Shot Question Generation that follows this intuitive process. In addition to the input KB triple, we feed our model with

---

<sup>7</sup>replicate the observation <http://bit.ly/2GvVHae>

a set of textual contexts paired with the input KB triple through distant supervision. Our model derives an encoder-decoder architecture, in which the encoder encodes the input KB triple, along with a set of textual contexts into hidden representations. Those hidden representations are fed to a decoder equipped with an attention mechanism to generate an output question.

In the Zero-Shot setup, the emergence of new predicates and new class types during test time requires new lexicalizations to express these predicates and classes in the output question. These lexicalizations might not be encountered by the model during training time and hence do not exist in the model vocabulary, or have been seen only few times not enough to learn a good representation for them by the model. Recent works on Text Generation tackle the rare words/unknown words problem using copy actions [102, 62]: words with a specific position are copied from the source text to the output text – although this process is blind to the role and nature of the word in the source text. Inspired by research in open information extraction [50] and structure-content neural language models [88], in which part-of-speech tags represent a distinctive feature when representing relations in text, we extend these positional copy actions. Instead of copying a word in a specific position in the source text, our model copies a word with a specific part-of-speech tag from the input text – we refer to those as part-of-speech copy actions. Experiments show that our model using contexts through distant supervision significantly outperforms the strongest baseline among six (+2.04 BLEU-4 score). Adding our copy action mechanism further increases this improvement (+2.39). Additionally, a human evaluation complements the comprehension of our model for edge cases; it supports the claim that the improvement brought by our copy action mechanism is even more significant than what the BLEU score suggests.

### 4.2.1 Related Work

QG became an essential component in many applications such as education [72], tutoring [59, 48] and dialogue systems [154]. Here, we focus on the problem of QG from structured KB and how we can generalize it to unseen predicates and entity types. [153] generate quiz questions from KB triples. Verbalization of entities and predicates relies on their existing labels in the KB and a dictionary. [152] use an encoder-decoder architecture with attention mechanism trained on the SimpleQuestions dataset [15]. [33] generate paraphrases of given questions to increase the performance of QA systems;

paraphrases are generated relying on paraphrase datasets, neural machine translation and rule mining. [85] generate a set of QA pairs given a KB entity. They model the problem of QG as a sequence-to-sequence problem by converting all the KB entities to a set of keywords. None of the previous work in QG from KB address the question of generalizing to unseen predicates and entity types.

Textual information has been used before in the Zero-Shot learning. [156] use information in pretrained word vectors for Zero-Shot visual object recognition. [96] incorporates a natural language question to the relation query to tackle Zero-Shot relation extraction problem.

Previous work in machine translation dealt with rare or unseen word problem problem for translating names and numbers in text. [102] propose a model that generates positional placeholders pointing to some words in source sentence and copy it to target sentence (*copy actions*). [62, 61] introduce separate trainable modules for copy actions to adapt to highly variable input sequences, for text summarization. For text generation from tables, [92] extend positional copy actions to copy values from fields in the given table. For QG, [152] use a placeholder for the subject entity in the question to generalize to unseen entities. Their work is limited to unseen entities and does not study how they can generalize to unseen predicates and entity types.

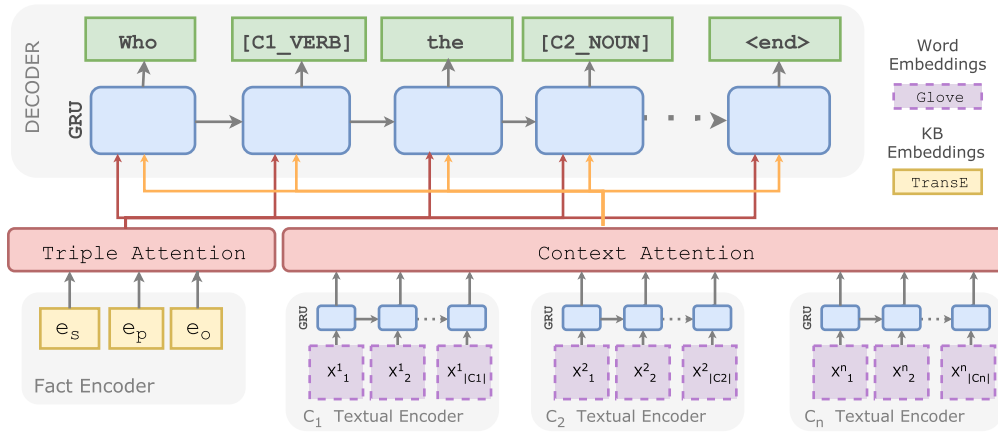


Figure 4.3: The proposed model for Question Generation. The model consists of a single fact encoder and  $n$  textual context encoders, each consists of a separate GRU. At each time step  $t$ , two attention vectors generated from the two attention modules are fed to the decoder to generate the next word in the output question.

### 4.2.2 A Model for Zero-Shot QG

Let  $F = \{s, p, o\}$  be the input fact provided to our model consisting of a subject  $s$ , a predicate  $p$  and an object  $o$ , and  $C$  be the set of textual contexts associated to this fact. Our goal is to learn a model that generates a sequence of  $T$  tokens  $Y = y_1, y_2, \dots, y_T$  representing a question about the subject  $s$ , where the object  $o$  is the correct answer. Our model approximates the conditional probability of the output question given an input fact  $p(Y|F)$ , to be the probability of the output question, given an input fact and the additional textual context  $C$ , modelled as follows:

$$p(Y|F) = \prod_{t=1}^T p(y_t|y_{<t}, F, C) \quad (4.2)$$

where  $y_{<t}$  represents all previously generated tokens until time step  $t$ . Additional textual contexts are natural language representation of the triples that can be drawn from a corpus – our model is generic to any textual contexts that can be additionally provided, though we describe in sub Section 4.2.4 how to create such texts from Wikipedia.

Our model derives the encoder-decoder architecture of [164, 5] with two encoding modules: a feed forward architecture encodes the input triple (sec. 4.2.2) and a set of recurrent neural network (RNN) to encode each textual context (sec. 4.2.3). Our model has two attention modules [5]: one acts over the input triple and another acts over the input textual contexts (sec. 4.2.3). The decoder (sec. 4.2.3) is another RNN that generates the output question. At each time step, the decoder chooses to output either a word from the vocabulary or a special token indicating a copy action (sec. 4.2.3) from any of the textual contexts.

#### Fact Encoder

Given an input fact  $F = \{s, p, o\}$ , let each of  $e_s$ ,  $e_p$  and  $e_o$  be a 1-hot vectors of size  $K$ . The fact encoder encodes each 1-hot vector into a fixed size vector  $h_s = \mathbf{E}_f e_s$ ,  $h_p = \mathbf{E}_f e_p$  and  $h_o = \mathbf{E}_f e_o$ , where  $\mathbf{E}_f \in \mathbb{R}^{H_k \times K}$  is the KB embedding matrix,  $H_k$  is the size of the KB embedding and  $K$  is the size of the KB vocabulary. The *encoded fact*  $h_f \in \mathbb{R}^{3H_k}$  represents the concatenation of those three vectors and we use it to initialize

the decoder.

$$h_f = [h_s; h_p; h_o] \quad (4.3)$$

$\mathbf{E}_f$  can be another parameter in the model to be learned. However, following [152] we learn this embedding matrix separately using *TransE* [15], a model for learning KB vector representations. We fix those weights and do not allow the encoder to update them during training time. Following [152], we learn  $\mathbf{E}_f$  using *TransE* [15]. We fix its weights and do not allow their update during training time.

### 4.2.3 Textual Context Encoder

Given a set of  $n$  textual contexts  $C = \{c_1, c_2, \dots, c_n : c_j = (x_1^j, x_2^j, \dots, x_{|c_j|}^j)\}$ , where  $x_i^j$  represents the 1-hot vector of the  $i^{\text{th}}$  token in the  $j^{\text{th}}$  textual context  $c_j$ , and  $|c_j|$  is the length of the  $j^{\text{th}}$  context. We use a set of  $n$  Gated Recurrent Neural Networks (GRU) [23] to encode each of the textual concepts separately:

$$h_i^{c_j} = GRU_j \left( \mathbf{E}_c x_i^j, h_{i-1}^{c_j} \right) \quad (4.4)$$

where  $h_i^{c_j} \in \mathbb{R}^{H_c}$  is the hidden state of the GRU that is equivalent to  $x_i^j$  and of size  $H_c$ .  $\mathbf{E}_c$  is the input word embedding matrix. The *encoded context* represents the encoding of all the textual contexts; it is calculated as the concatenation of all the final states of all the encoded contexts:

$$h_c = [h_{|c_1|}^{c_1}; h_{|c_2|}^{c_2}; \dots; h_{|c_n|}^{c_n}]. \quad (4.5)$$

### Decoder

For the decoder we use another GRU with an attention mechanism [5], in which the decoder hidden state  $s_t \in \mathbb{R}^{H_d}$  at each time step  $t$  is calculated as:

$$s_t = z_t \circ s_{t-1} + (1 - z_t) \circ \tilde{s}_t, \quad (4.6)$$

Where:

$$\tilde{s}_t = \tanh \left( W E_w y_{t-1} + U [r_t \circ s_{t-1}] + A [a_t^f; a_t^c] \right) \quad (4.7)$$

$$z_t = \sigma \left( W_z E_w y_{t-1} + U_z s_{t-1} + A_z [a_t^f; a_t^c] \right) \quad (4.8)$$

$$r_t = \sigma \left( W_r E_w y_{t-1} + U_r s_{t-1} + A_r [a_t^f; a_t^c] \right) \quad (4.9)$$

$W, W_z, W_r \in \mathbb{R}^{m \times H_d}$ ,  $U, U_z, U_r, A, A_z, A_r \in \mathbb{R}^{H_d \times H_d}$  are learnable parameters of the GRU.  $E_w \in \mathbb{R}^{m \times V}$  is the word embedding matrix,  $m$  is the word embedding size and  $H_d$  is the size of the decoder hidden state.  $a_t^f, a_t^c$  are the outputs of the fact attention and the context attention modules respectively, detailed in the following subsection.

In order to enforce the model to pair output words with words from the textual inputs, we couple the word embedding matrices of both the decoder  $E_w$  and the textual context encoder  $E_c$  (eq.(4.4)). We initialize them with GloVe embeddings [132] and allow the network to tune them.

The first hidden state of the decoder  $s_0 = [h_f; h_c]$  is initialized using a concatenation of the encoded fact (eq.(4.3)) and the encoded context (eq.(4.5)) .

At each time step  $t$ , after calculating the hidden state of the decoder, the conditional probability distribution over each token  $y_t$  of the generated question is computed as the  $\text{softmax}(W_o s_t)$  over all the entries in the output vocabulary,  $W_o \in \mathbb{R}^{H_d \times V}$  is the weight matrix of the output layer of the decoder.

### Attention

Our model has two attention modules:

**Triple attention** over the input triple to determine at each time step  $t$  an attention-based encoding of the input fact  $a_t^f \in \mathbb{R}^{H_k}$ :

$$a_t^f = \alpha_{s,t} h_s + \alpha_{p,t} h_p + \alpha_{o,t} h_o \quad , \quad (4.10)$$

$\alpha_{s,t}, \alpha_{p,t}, \alpha_{o,t}$  are scalar values calculated by the attention mechanism to determine at each time step which of the encoded subject, predicate, or object the decoder should attend to.

**Textual contexts attention** over all the hidden states of all the textual contexts  $a_t^c \in$

$\mathbb{R}^{H_c}$ :

$$a_t^c = \sum_{i=1}^{|C|} \sum_{j=1}^{|c_i|} \alpha_{t,j}^{c_i} h_j^{c_i}, \quad (4.11)$$

$\alpha_{t,j}^{c_i}$  is a scalar value determining the weight of the  $j^{\text{th}}$  word in the  $i^{\text{th}}$  context  $c^i$  at time step  $t$ .

Given a set of encoded input vectors  $I = \{h_1, h_2, \dots, h_k\}$  and the decoder previous hidden state  $s_{t-1}$ , the attention mechanism calculates  $\alpha_t = \alpha_{i,t}, \dots, \alpha_{k,t}$  as a vector of scalar weights, each  $\alpha_{i,t}$  determines the weight of its corresponding encoded input vector  $h_i$ .

$$e_{i,t} = \mathbf{v}_a^\top \tanh(\mathbf{W}_a \mathbf{s}_{t-1} + \mathbf{U}_a \mathbf{h}_i) \quad (4.12)$$

$$\alpha_{i,t} = \frac{\exp(e_{i,t})}{\sum_{j=1}^k \exp(e_{j,t})}, \quad (4.13)$$

where  $\mathbf{v}_a, \mathbf{W}_a, \mathbf{U}_a$  are trainable weight matrices of the attention modules. It is important to notice here that we encode each textual context separately using a different GRU, but we calculate an overall attention over all tokens in all textual contexts: at each time step the decoder should ideally attend to only one word from all the input contexts.

	What caused the [C1_NOUN] of the [C3_NOUN] [S] ?
C1	[S] <b>death</b> by [O] [S] [ <b>C1_NOUN</b> ] [C1_ADP] [O]
C2	Disease [C2_NOUN]
C3	Musical <b>artist</b> [C3_ADJ] [ <b>C3_NOUN</b> ]

Table 4.7: An annotated example of part-of-speech copy actions from several input textual contexts (C1, C2, C3), the words or placeholders in bold are copied in the generated question

### Part-Of-Speech Copy Actions

We use the method of [102] by modeling all the copy actions on the data level through an annotation scheme. This method treats the model as a black box, which makes it adaptable to any text generation model. Instead of using positional copy actions, we use the part-of-speech information to decide the alignment process between the input and output texts to the model. Each word in every input textual context is replaced by a special token containing a combination of its context id (e.g. C1) and its POS tag (e.g. NOUN). Then, if a word in the output question matches a word in a textual context, it is replaced with its corresponding tag as shown in Table 4.7.

Unlike [152, 92] we model the copy actions in the input and the output levels. Our model does not have the drawback of losing the semantic information when replacing words with generic placeholders, since we provide the model with the input triple through the fact encoder. During inference the model chooses to either output words from the vocabulary or special tokens to copy from the textual contexts. In a post-processing step those special tokens are replaced with their original words from the textual contexts.

#### 4.2.4 Textual contexts dataset

As a source of question paired with KB triples we use the SimpleQuestions dataset [15]. It consists of 100K questions with their corresponding triples from Freebase, and was created manually through crowdsourcing. When asked to form a question from an input triple, human annotators usually tend to mainly focus on expressing the predicate of the input triple. For example, given a triple with the predicate `fb:car/manufacturer` the user may ask "What is the manufacturer of [S] ?". Annotators may specify the entity type of the subject or the object of the triple: "What is the manufacturer of the *spacecraft* [S]?" or "Which *company* manufactures [S]?". Motivated by this example we chose to associate each input triple with three textual contexts of three different types. The first is a phrase containing lexicalization of the predicate of the triple. The second and the third are two phrases containing the entity type of the subject and the object of the triple. In what follows we show the process of collection and preprocessing of those textual contexts.



### Collection of Textual Contexts

We extend the set of triples given in the SimpleQuestions dataset by using the FB5M [15] subset of Freebase. As a source of text documents, we rely on Wikipedia articles.

**Predicate textual contexts:** In order to collect textual contexts associated with the SimpleQuestions triples, we follow the distant supervision setup for relation extraction [114].

First, we reuse the *T-REx* framework to align each triple in the FB5M KB to sentences in Wikipedia if the subject and the object of this triple co-occur in the same sentence. We use a simple string matching heuristic to find entity mentions in text<sup>8</sup>. Afterwards we reduce the sentence to the set of words that appear on the dependency path between the subject and the object mentions in the sentence. We replace the positions of the subject and the object mentions with [S] and [O] to keep track of the information about the direction of the relation. The top occurring patterns for each predicate is associated to this predicate as its textual context. Table 4.8 shows examples of predicates and their corresponding textual context.

**Sub-Type and Obj-Type textual contexts:** We use the labels of the entity types as the sub-type and obj-type textual contexts. We collect the list of entity types of each entity in the FB5M through the predicate `fb:type/instance`. If an entity has multiple entity types we pick the entity type that is mentioned the most in the first sentence of each Wikipedia article. Thus the textual contexts will opt for entity types that is more natural to appear in free text and therefore questions.

### Generation of Special tokens

To generate the special tokens for copy actions (sec. 4.2.3) we run POS tagging on each of the input textual contexts<sup>9</sup>. We replace every word in each textual context with a combination of its context id (e.g. C1) and its POS tag (e.g. NOUN). If the same POS tag appears multiple times in the textual context, it is given an additional id (e.g.

---

<sup>8</sup> We map Freebase entities to Wikidata through the Wikidata property P646, then we extract their labels and aliases. We use the Wikidata truthy dump: <https://dumps.wikimedia.org/wikidatawiki/entities/>

<sup>9</sup>For the predicate textual contexts we run pos tagging on the original text not the lexicalized dependency path

Freebase Relation	Predicate Textual Context
person/place_of_birth	[O] is birthplace of [S]
currency/former_countries	[S] was currency of [O]
dish/cuisine	[O] dish [S]
airliner_accident/flight_origin	[S] was flight from [O]
film_featured_song/performer	[S] is release by [O]
airline_accident/operator	[S] was accident for [O]
genre/artists	[S] became a genre of [O]
risk_factor/diseases	[S] increases likelihood of [O]
book/illustrations_by	[S] illustrated by [O]
religious_text/religion	[S] contains principles of [O]
spacecraft/manufacturer	[S] spacecraft developed by [O]

Table 4.8: Table showing an example of textual contexts extracted for freebase predicates

C1\_NOUN\_2). If a word in the output question overlaps with a word in the input textual context, this word is replaced by its corresponding tag.

For sentence and word tokenization we use the Regex tokenizer from the NLTK toolkit [12], and for POS tagging and dependency parsing we use the Spacy<sup>10</sup> implementation.

## 4.2.5 Experiments

### Zero-Shot Setups

We develop three setups that follow the same procedure as [96] for Zero-Shot relation extraction to evaluate how our model generalizes to: 1) unseen predicates, 2) unseen sub-types and 3) unseen obj-types.

For the unseen predicates setup we group all the samples in SimpleQuestions by the predicate of the input triple, and keep groups that contain at least 50 samples. Afterwards we randomly split those groups to 70% train, 10% valid and 20% test mutual exclusive sets respectively. This guarantees that if the predicate `fb:person/place_of_birth` for example shows during test time, the training and validation set will not contain any input triples having this predicate. We repeat this process to create 10 cross validation folds, in our evaluation we report the mean and standard deviation results across those 10 folds. While doing this we make sure that the number of samples in each fold – not

<sup>10</sup><https://spacy.io/>

	<b>Train</b>	<b>Valid</b>	<b>Test</b>	
<b>pred</b>	# pred	169.4	24.2	48.4
	# samples	55566.7	7938.1	15876.2
	% samples	70.0 $\pm$ 2.77	10.0 $\pm$ 1.236	20.0 $\pm$ 2.12
<b>sub-types</b>	# types	112.7	16.1	32.2
	# samples	60002.6	8571.8	17143.6
	% samples	70.0 $\pm$ 7.9	10.0 $\pm$ 3.6	20.0 $\pm$ 6.2
<b>obj-types</b>	# types	521.6	189.9	282.2
	# samples	57878.1	8268.3	16536.6
	% samples	70.0 $\pm$ 4.7	10.0 $\pm$ 2.5	20.0 $\pm$ 3.8

Table 4.9: Dataset statistics across 10 folds for each experiment

only unique predicates – follow the same 70%, 30%, 10% distribution. We repeat the same process for the subject entity types and object entity types (answer types) individually. Similarly, for example in the unseen object-type setup, the question “*Which artist was born in Berlin?*” appearing in the test set means that, there is no question in the training set having an entity of type *artist*. Table 4.9 shows the mean number of samples, predicates, sub-types and obj-types across the 10 folds for each experiment setup.

### Baselines

**1) SELECT:** is a baseline built from [152] and adapted for the zero shot setup. During test time given a fact  $F$ , this baseline picks a fact  $F_c$  from the training set and outputs the question that corresponds to it. For evaluating unseen predicates,  $F_c$  has the same answer type (obj-type) as  $F$ . And while evaluating unseen sub-types or obj-types,  $F_c$  and  $F$  have the same predicate.

**2) R-TRANSE:** is an extension that we propose for SELECT. The input triple is encoded using the concatenation of the TransE embeddings of the subject, predicate and object. At test time, R-TRANSE picks a fact from the training set that is the closest to the input fact using cosine similarity and outputs the question that corresponds to it. We provide two versions of this baseline: **R-TRANSE** which indexes and retrieves raw questions with only a single placeholder for the subject label, such as in [152]. And

	Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE <sub>L</sub>	METEOR
Unseen Predicates	SELECT	46.81 ± 2.12	38.62 ± 1.78	31.26 ± 1.9	23.66 ± 2.22	52.04 ± 1.43	27.11 ± 0.74
	IR	48.43 ± 1.64	39.13 ± 1.34	31.4 ± 1.66	23.59 ± 2.36	52.88 ± 1.24	27.34 ± 0.55
	IR <sub>COPY</sub>	48.22 ± 1.84	38.82 ± 1.5	31.01 ± 1.72	23.12 ± 2.24	52.72 ± 1.26	27.24 ± 0.57
	R-TRANSE	49.09 ± 1.69	40.75 ± 1.42	33.4 ± 1.7	25.97 ± 2.22	54.07 ± 1.31	28.13 ± 0.54
	R-TRANSE <sub>COPY</sub>	49.0 ± 1.76	40.63 ± 1.48	33.28 ± 1.74	25.87 ± 2.23	54.09 ± 1.35	28.12 ± 0.57
	Encoder-Decoder	58.92 ± 2.05	47.7 ± 1.62	38.18 ± 1.86	28.71 ± 2.35	59.12 ± 1.16	34.28 ± 0.54
	Our-Model	60.8 ± 1.52	49.8 ± 1.37	40.32 ± 1.92	30.76 ± 2.7	60.07 ± 0.9	35.34 ± 0.43
	Our-Model <sub>copy</sub>	<b>62.44</b> ± 1.85	<b>50.62</b> ± 1.46	<b>40.82</b> ± 1.77	<b>31.1</b> ± 2.46	<b>61.23</b> ± 1.2	<b>36.24</b> ± 0.65

Table 4.10: Evaluation results of our model and all other baselines for the unseen predicate evaluation setup

**R-TRANSE<sub>copy</sub>** which indexes and retrieves questions using our copy actions mechanism (sec. 4.2.3).

**3) IR:** is an information retrieval baseline. Information retrieval has been used before as baseline for QG from text input [145, 36]. We rely on the textual context of each input triple as the search keyword for retrieval. First, the IR baseline encodes each question in the training set as a vector of TF-IDF weights [78] and then does dimensionality reduction through LSA [66]. At test time the textual context of the input triple is converted into a dense vector using the same process and then the question with the closest cosine distance to the input is retrieved. We provide two versions of this baseline: **IR** on raw text and **IR<sub>copy</sub>** on text with our placeholders for copy actions.

**4) Encoder-Decoder.** Finally, we compare our model to the Encoder-Decoder model with a single placeholder, the best performing model from [152]. We initialize the encoder with TransE embeddings and the decoder with GloVe word embeddings. Although this model was not originally built to generalize to unseen predicates and entity types, it has some generalization abilities represented in the encoded information in the pre-trained embeddings. Pretrained KB terms and word embeddings encode relations between entities or between words as translations in the vector space. Thus the model might be able to map new classes or predicates in the input fact to new words in the output question.

	<b>Model</b>	<b>BLEU-4</b>	<b>ROUGE<sub>L</sub></b>
<b>Sub-Types</b>	R-TRANSE	32.41 $\pm$ 1.74	59.27 $\pm$ 0.92
	Encoder-Decoder	42.14 $\pm$ 2.05	68.95 $\pm$ 0.86
	Our-Model	42.13 $\pm$ 1.88	69.35 $\pm$ 0.9
	Our-Model <sub>copy</sub>	<b>42.2</b> $\pm$ 2.0	<b>69.37</b> $\pm$ 1.0
<b>Obj-Types</b>	R-TRANSE	30.59 $\pm$ 1.3	57.37 $\pm$ 1.17
	Encoder-Decoder	37.79 $\pm$ 2.65	65.69 $\pm$ 2.25
	Our-Model	37.78 $\pm$ 2.02	65.51 $\pm$ 1.56
	Our-Model <sub>copy</sub>	<b>38.02</b> $\pm$ 1.9	<b>66.24</b> $\pm$ 1.38

Table 4.11: Automatic evaluation of our model against selected baselines for unseen sub-types and obj-types

### Training & Implementation Details

To train the neural network models we optimize the negative log-likelihood of the training data with respect to all the model parameters. For that we use the RMSProp optimization algorithm with a decreasing learning rate of 0.001, mini-batch size = 200, and clipping gradients with norms larger than 0.1. We use the same vocabulary for both the textual context encoders and the decoder outputs. We limit our vocabulary to the top 30,000 words including the special tokens. For the word embeddings we chose GloVe [132] pretrained embeddings of size 100. We train TransE embeddings of size  $H_k = 200$ , on the FB5M dataset [15] using the TransE model implementation from [100]. We set GRU hidden size of the decoder to  $H_d = 500$ , and textual encoder to  $H_c = 200$ . The networks hyperparameters are set with respect to the final BLEU-4 score over the validation set. All neural networks are implemented using Tensorflow [2]. All experiments and models source code are publicly available<sup>11</sup> for the sake of reproducibility.

### Automatic Evaluation Metrics

To evaluate the quality of the generated question, we compare the original labeled questions by human annotators to the ones generated by each variation of our model and the baselines. We rely on a set of well established evaluation metrics for text generation:

<sup>11</sup><https://github.com/hadyelsahar/Zeroshot-QuestionGeneration>

BLEU-1, BLEU-2, BLEU-3, BLEU-4 [130], METEOR [31] and ROUGE<sub>L</sub> [98] (refer to chapter 2 for more information about metrics for NLG evaluation).

### Human Evaluation

Automatic Metrics for evaluating text generation such as BLEU and METEOR give a measure of how close the generated questions are to the target correct labels. However, they still suffer from many limitations [128]. Automatic metrics might not be able to evaluate directly whether a specific predicate was explicitly mentioned in the generated text or not.

As an example, taking a target question and two corresponding generated questions *A* and *B*:

What kind of film is kill bill vol. 2?	BLEU
A) What is <i>the name of</i> the film kill bill vol. 2?	71
B) Which <b>genre</b> is kill bill vol. 2 in?	55

We can find that the sentence *A* having a better BLEU score than *B* although it is not able to express the correct target predicate (*film genre*). For that reason we decide to run two further human evaluations to directly measure the following:

**Predicate identification:** annotators were asked to indicate whether the generated question contains the given predicate in the fact or not, either directly or implicitly.

**Naturalness:** following [122], we measure the comprehensibility and readability of the generated questions. Each annotator was asked to rate each generated question using a scale from 1 to 5, where: (5) perfectly clear and natural, (3) artificial but understandable, and (1) completely not understandable. We run our studies on 100 randomly sampled input facts alongside with their corresponding generated questions by each of the systems using the help of 4 annotators. We compare the Encoder-decoder model by [152] the best performing baseline to three variations of our model: 1) without copy actions 2) with copy actions from the subject type and object type textual context 3) with copy actions from all textual contexts.

<b>Model</b>	<b>% Pred. Identified</b>	<b>Natural.</b>
Encoder-Decoder	6	3.14
Our-Model (No Copy)	6	2.72
Our-Model <sub>copy</sub> (Types context)	<b>37</b>	<b>3.21</b>
Our-Model <sub>copy</sub> (All contexts)	<b>46</b>	2.61

Table 4.12: results of Human evaluation on % of predicates identified and naturalness 0-5

#### 4.2.6 Results & Discussion

**Automatic Evaluation** Table 4.10 shows results of our model compared to all other baselines across all evaluation metrics. Our that encodes the KB fact and textual contexts achieves a significant enhancement over all the baselines in all evaluation metrics, with +2.04 BLEU-4 score than the Encoder-Decoder baseline. Incorporating the part-of-speech copy actions further improves this enhancement to reach +2.39 BLEU-4 points. Among all baselines, the Encoder-Decoder baseline and the R-TRANSE baseline performed the best. This shows that TransE embeddings encode intra-predicates information and intra-class-types information to a great extent, and can generalize to some extent to unseen predicates and class types.

Similar patterns can be seen in the evaluation on unseen sub-types and obj-types (Table 4.11). Our model with copy actions was able to outperform all the other systems. Majority of systems have reported a significantly higher BLEU-4 scores in these two tasks than when generalizing to unseen predicates (+12 and +8 BLEU-4 points respectively). This indicates that these tasks are relatively easier and hence our models achieve relatively smaller enhancements over the baselines.

**Human Evaluation** Table 4.12 shows how different variations of our system can express the unseen predicate in the target question with comparison to the Encoder-Decoder baseline.

Our proposed copy actions have scored a significant enhancement in the identification of unseen predicates with up to +40% more than best performing baseline and our model version without the copy actions.

By examining some of the generated questions (Table 4.13) we see that models without

copy actions can generalize to unseen predicates that only have a very similar free-base predicate in the training set. For example `fb:tv_program/language` and `fb:film/language`, if one of those predicates exists in the training set the model can use the same questions for the other during test time.

Copy actions from the sub-type and the obj-type textual contexts can generalize to a great extent to unseen predicates because of the overlap between the predicate and the object type in many questions (Example 2 Table 4.13). Adding the predicate context to our model has enhanced model performance for expressing unseen predicates by +9% (Table 4.12). However we can see that it has affected the naturalness of the question. The post processing step does not take into consideration that some verbs and prepositions do not fit in the sentence structure, or that some words are already existing in the question words (Example 4 Table 4.13). This does not happen as much when having copy actions from the sub-type and the obj-type contexts because they are mainly formed of nouns which are more interchangeable than verbs or prepositions. A post-processing step to reform the question instead of direct copying from the input source is considered in our future work.

#### 4.2.7 Conclusion

We present a new neural model for question generation from knowledge bases, with a main focus on predicates, subject types or object types that were not seen at the training phase (Zero-Shot Question Generation). Our model is based on an encoder-decoder architecture that leverages textual contexts of triples, two attention layers for triples and textual contexts and finally a part-of-speech copy action mechanism. Our method exhibits significantly better results for Zero-Shot QG than a set of strong baselines including the state-of-the-art question generation from KB. Additionally, a complimentary human evaluation, helps in showing that the improvement brought by our part-of-speech copy action mechanism is even more significant than what the automatic evaluation suggests. The source code and the collected textual contexts are provided for the community<sup>12</sup>

---

<sup>12</sup><https://github.com/hadyelsahar/Zeroshot-QuestionGeneration>



1	<b>Reference</b>	<b>what language is spoken in the tv show three sheets?</b>
	<b>Enc-Dec.</b>	in what <b>language</b> is three sheets in?
	<b>Our-Model</b>	what the the player is the three sheets?
	<b>Our-Model<sub>Copy</sub></b>	what is the <b>language</b> of three sheets?
2	<b>Reference</b>	<b>how is roosevelt in Africa classified?</b>
	<b>Enc-Dec.</b>	what is the name of a roosevelt in Africa?
	<b>Our-Model</b>	what is the name of the movie roosevelt in Africa?
	<b>Our-Model<sub>Copy</sub></b>	what is a <b>genre</b> of roosevelt in Africa?
3	<b>Reference</b>	<b>where can 5260 philvéron be found?</b>
	<b>Enc-Dec.</b>	what is a release some that 5260 philvéron wrote?
	<b>Our-Model</b>	what is the name of an artist 5260 philvéron?
	<b>Our-Model<sub>Copy</sub></b>	which <b>star system</b> contains the star system body 5260 philvéron?
4	<b>Reference</b>	<b>which university did ezra cornell create?</b>
	<b>Enc-Dec.</b>	which films are part of ezra cornell?
	<b>Our-Model</b>	what is a position of ezra cornell?
	<b>Our-Model<sub>Copy</sub></b>	what <i>founded</i> the name of a university that ezra cornell <b>founded</b> ?
5	<b>Reference</b>	<b>who founded snocap , inc .?</b>
	<b>Enc-Dec.</b>	which asian snocap is most as?
	<b>Our model</b>	what is the name of a person of snocap?
	<b>Our-Model<sub>Copy</sub></b>	who is the <b>person behind</b> snocap?
6	<b>Reference</b>	<b>which 1992 album was produced by daniel barenboim?</b>
	<b>Enc-Dec.</b>	which german contains daniel barenboim ?
	<b>Our model</b>	which <b>album</b> is from the subject daniel barenboim
	<b>+ Copy</b>	what was the name of <i>a album</i> that is daniel barenboim ?

Table 4.13: Examples of generated questions from different systems in comparison

## Chapter 5

# Limitations of Answer Display

*This chapter is an equal contribution between Lucie-Aimée Kaffee, Pavlos Vougiouklis and myself.*

### 5.1 Learning to Generate Wikipedia Summaries for Under-served Languages from Wikidata

Despite the fact that Wikipedia exists in 287 languages, the existing content is unevenly distributed. The content of the most under-resourced Wikipedias is maintained by a limited number of editors – they cannot curate the same volume of articles as the editors of large Wikipedia language-specific communities. It is therefore of the utmost social and cultural interests to address languages for which native speakers have only access to an impoverished Wikipedia. We propose an automatic approach to generate textual summaries that can be used as a starting point for the editors of the involved Wikipedias. We propose an end-to-end trainable model that generates a textual summary given a set of KB triples as input. We apply our model on two languages that have a severe lack of both editors and articles on Wikipedia. First, Esperanto is an easily acquired artificially created language which makes it less data needy and a more suitable starting point for exploring the challenges of this task. Second, Arabic is a morphologically rich language that is much more challenging to work, mainly due to its significantly larger vocabulary. As shown in Table 5.1 both Arabic and Esperanto suffer from a severe lack of content and active editors compared to the English Wikipedia which is currently the biggest one

in terms of number of articles. Both Arabic and Esperanto suffer a severe lack of content (Arabic with 541,166 and Esperanto 241,901 articles, compared to 5,483,928 in the English Wikipedia) and active editors (2,849 and 7,818 active users respectively) compared to the English Wikipedia (129,237 active users). Our research is mostly related to previous work on adapting the general encoder-decoder framework for the generation of Wikipedia summaries [92, 22, 175]. Nonetheless, all these approaches focus on the task of biographies generation, and only in English – the language with the most language resources and knowledge bases available. In contrast with these works, we explore the generation of sentences in an open-domain, multilingual context. The model from [92] takes the Wikipedia infobox as an input, while [22] uses a sequence of slot-value pairs extracted from Wikidata. Both models are only able to generate single-subject relationships. In our model the input triples go beyond the single-subject relationships of a Wikipedia infobox or a Wikidata page about a specific item (subsection 5.1.1). During test time, the model may encounter entities that it has either not seen enough or not seen at all during training.

While [175] tackles this issue by using a set of triples as input, which are limited to instance-type-related information leveraged from DBpedia. Similarly to our approach, the model proposed by [175] accepts a set of triples as input, however, it leverages instance-type-related information from DBpedia in order to generate text that addresses rare or unseen entities. Our solution is much broader since it does not rely on the assumption that unseen triples will adopt the same pattern of properties and entities' instance types pairs as the ones that have been used for training. To this end, we use copy actions over the labels of entities in the input triples. This relates to previous works in machine translation which deals with rare or unseen word problem for translating names and numbers in text. [102] propose a model that generates positional placeholders pointing to some words in source sentence and copy it to target sentence (*copy actions*). [63] introduce separate trainable modules for copy actions to adapt to highly variable input sequences, for text summarisation. For text generation from tables, [92] extend positional copy actions to copy values from fields in the given table. For Question Generation, [152] use a placeholder for the subject entity in the question to generalize to unseen entities.

We evaluate our approach by measuring how close our synthesised summaries can be to actual summaries in Wikipedia against two other baselines of different natures:

	Arabic	Esperanto	English
# of Articles	541,166	241,901	5,483,928
# of Active Users	7,818	2,849	129,237
Vocab. Size	2.2M	1.5M	2.0M

Table 5.1: Recent page statistics and number of unique words (vocab. size) of Esperanto, Arabic and English Wikipedias.

<b>Triples</b>	Q490900 (Florida)	P31 (estas)	Q747074 (komunumo de Italio)
	Q490900 (Florida)	P17 (ŝtato)	Q38 (Italio)
	Q30025755 (Florida)	P1376 (ĉefurbo de)	Q490900 (Florida)
<b>Textual Summary</b>	Florida estas komunumo de Italio.		
<b>Vocab. Extended</b>	[[Q490900, Florida]] estas komunumo de [[P17]].		

Table 5.2: Training example: a set of triples about *Florida*. Subsequently, our system summarises the input set in the form of text. The vocabulary extended summary is the one on which we train our model.

a language model, and an information retrieval template-based solution. Our model substantially outperforms all the baselines in all evaluation metrics in both Esperanto and Arabic. In this work we present the following contributions: i) We investigate the task of generating textual summaries from Wikidata triples in underserved Wikipedia languages across multiple domains, and ii) We use an end-to-end model with copy actions adapted to this task. Our datasets, results, and experiments are available at: <https://github.com/pvougou/Wikidata2Wikipedia>.

### 5.1.1 Model

Our approach is inspired by similar encoder-decoder architectures that have already been employed on similar text generative tasks [152, 175].

#### Encoding the Triples

The encoder part of the model is a feed-forward architecture that encodes the set of input triples into a fixed dimensionality vector, which is subsequently used to initialise the decoder. Given a set of un-ordered triples  $F_E = \{f_1, f_2, \dots, f_R : f_j = (s_j, p_j, o_j)\}$ , where  $s_j$ ,  $p_j$  and  $o_j$  are the one-hot vector representations of the respective subject,

property and object of the  $j$ -th triple, we compute an embedding  $h_{f_j}$  for the  $j$ -th triple by forward propagating as follows:

$$h_{f_j} = q(\mathbf{W}_h[\mathbf{W}_{in}s_j; \mathbf{W}_{in}p_j; \mathbf{W}_{in}o_j]) , \quad (5.1)$$

$$h_{F_E} = \mathbf{W}_F[h_{f_1}; \dots; h_{f_{R-1}}; h_{f_R}] , \quad (5.2)$$

where  $h_{f_j}$  is the embedding vector of each triple  $f_j$ ,  $h_{F_E}$  is a fixed-length vector representation for all the input triples  $F_E$ .  $q$  is a non-linear activation function,  $[\dots; \dots]$  represents vector concatenation.  $\mathbf{W}_{in}, \mathbf{W}_h, \mathbf{W}_F$  are trainable weight matrices. Unlike [22], our encoder is agnostic with respect to the order of input triples. As a result, the order of a particular triple  $f_j$  in the triples set does not change its significance towards the computation of the vector representation of the whole triples set,  $h_{F_E}$ .

### Decoding the Summary

The decoder part of the architecture is a multi-layer RNN [23] with Gated Recurrent Units which generates the textual summary one token at a time. The hidden unit of the GRU at the first layer is initialised with  $h_{F_E}$ . At each timestep  $t$ , the hidden state of the GRU is calculated as follows:

$$h_t^l = \text{GRU}(h_{t-1}^l, h_t^{l-1}) \quad (5.3)$$

The conditional probability distribution over each token  $y_t$  of the summary at each timestep  $t$  is computed as the softmax( $\mathbf{W}_{out}h_t^L$ ) over all the possible entries in the summaries dictionary, where  $h_t^L$  is the hidden state of the last layer and  $\mathbf{W}_{out}$  is a bi-ased trainable weight matrix.

A summary consists of words and mentions of entity in the text. We adapt the concept of *surface form tuples* [175] in order to be able to learn an arbitrary number of different lexicalisations of the same entity in the summary (e.g. “aktorino”, “aktoro”). Figure 5.3 shows the architecture of our generative model when it is provided with the three triples of the idealised example of Table 5.2.

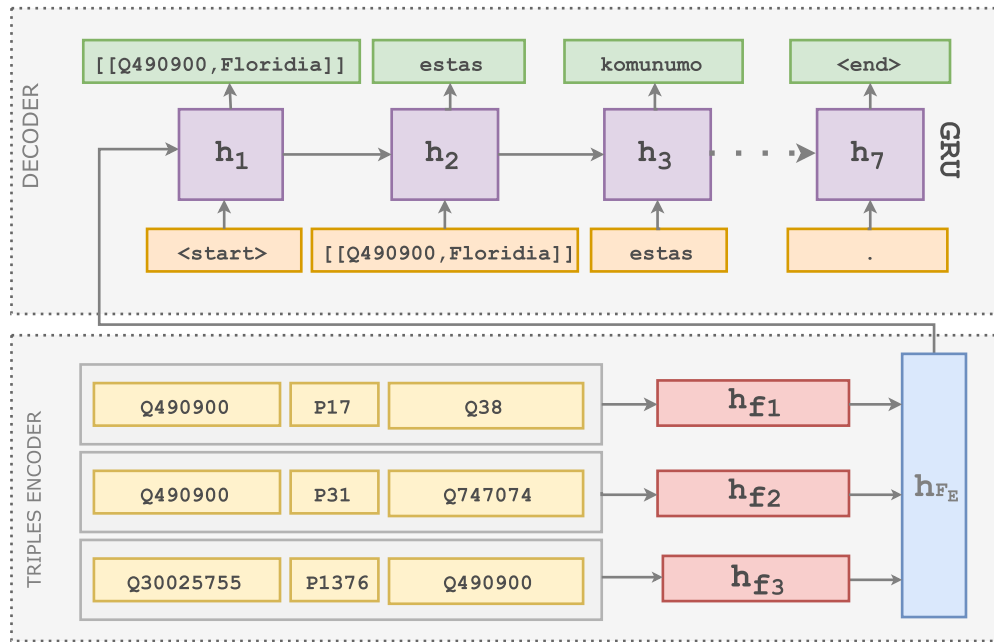


Figure 5.1: Model Overview

### Copy Actions

Following [102, 92] we model all the copy actions on the data level through a set of special tokens added to the basic vocabulary. Rare entities identified in text and existing in the input triples are being replaced by the token of the property of the relationship to which it was matched. We refer to those tokens as *property placeholders*. In Table 5.2,  $[[P17]]$  in the vocabulary extended summary is an example of property placeholder – would it be generated by our model, it is replaced with the label of the object of the triple with which they share the same property (i.e.  $Q490900$  (Florida)  $P17$  ( $\$$ stato)  $Q38$  (Italia)). When all the tokens of the summary are sampled, each property placeholder that is generated is mapped to the triple with which it shares the same property and is subsequently replaced with the textual label of the entity. We randomly choose an entity, in case there are more than one triple with the same property in the input triples set.

### 5.1.2 Implementation and Training Details

We implemented our neural network models using the Torch<sup>1</sup> package.

<sup>1</sup>Torch is a scientific computing package for Lua. It is based on the LuaJIT package.

We included the 15, 000 and 25, 000 most frequent tokens (i.e. either words or entities) of the summaries in Esperanto and Arabic respectively for target vocabulary of the textual summaries. Using a larger size of target dictionary in Arabic is due to its greater linguistic variability – Arabic vocabulary is 47% larger than Esperanto vocabulary (cf. Table 5.1). We replaced any rare entities in the text that participate in relations in the aligned triples set with the corresponding property placeholder of the upheld relations. We include all property placeholders that occur at least 20 times in each training dataset. Subsequently, the dictionaries of the Esperanto and Arabic summaries are expanded by 80 and 113 property placeholders respectively. In case the rare entity is not matched to any subject or object of the set of corresponding triples it is replaced by the special `<resource>` token. Each summary is augmented with a start-of-summary `<start>` and end-of-summary `<end>` tokens.

For the decoder, we use 1 layer of GRUs. We set the dimensionality of the decoder’s hidden state to 500 in Esperanto and 700 in Arabic. We initialise all parameters with random uniform distribution between  $-0.001$  and  $0.001$ , and we use Batch Normalisation before each non-linear activation function and after each fully-connected layer [76] on the encoder side [175] During training, the model tries to learn those parameters that minimise the sum of the negative log-likelihoods of a set of predicted summaries. The networks are trained using mini-batch of size 85. The weights are updated using Adam [87] (i.e. it was found to work better than Stochastic Gradient Descent, RMSProp and AdaGrad) with a learning rate of  $10^{-5}$ . An  $l_2$  regularisation term of 0.1 over each network’s parameters is also included in the cost function.

The networks converge after the 9th epoch in the Esperanto case and after the 11th in the Arabic case. During evaluation and testing, we do beam search with a beam size of 20, and we retain only the summary with the highest probability. We found that increasing the beam size resulted not only in minor improvements in terms of performance but also in a greater number of fully-completed generated summaries (i.e. summaries for which the special end-of-summary `<end>` token is generated).

### 5.1.3 Dataset

In order to train our models to generate summaries from Wikidata triples, we introduced a new dataset for text generation from KB triples in a multilingual setting and align it with the triples of its corresponding Wikidata Item. For each Wikipedia article, we

	Arabic	Esperanto
Avg. # of Tokens per Summary	28.1 ( $\pm 28.8$ )	26.4 ( $\pm 22.7$ )
Avg. # of Triples per Summary	8.1 ( $\pm 11.2$ )	11.0 ( $\pm 13.8$ )
Avg. # of Linked Named Entities	2.2 ( $\pm 1.0$ )	2.4 ( $\pm 1.1$ )
Avg. # of Aligned Triples	0.1 ( $\pm 0.4$ )	0.2 ( $\pm 0.5$ )
Vocabulary Size	344,827	226,447
Total # of Summaries	255,741	126,714

Table 5.3: Dataset statistics in Arabic and Esperanto.

extract and tokenise the first introductory sentence and align it with triples where its corresponding item appears as a subject or an object in the Wikidata truthy dump. In order to create the *surface form tuples* (i.e. subsection 5.1.1), we identify occurrences of entities in the text along with their verbalisations. We rely on keyword matching against labels from Wikidata expanded by the global language fallback chain introduced by Wikimedia<sup>2</sup> to overcome the lack of non-English labels in Wikidata [84].

For the *property placeholders*, we use the distant supervision assumption for relation extraction [114]. Entities that participate in relations with the main entity of the article are being replaced with their corresponding property placeholder tag. Table 5.3 shows statistics on the two corpora that we used for the training of our systems.

#### 5.1.4 Baselines

To demonstrate the effectiveness of our approach, we compare it to two competitive systems.

**KN** is a 5-gram Kneser-Ney (KN) [70] language model. KN has been used before as a baseline for text generation from structured data [92] and provided competitive results on a single domain in English. We also introduce a second KN model ( $\text{KN}_{\text{ext}}$ ), which is trained on summaries with the special tokens for copy actions. During test time, we use beam search of size 10 to sample from the learned language model.

**IR** is an Information Retrieval (IR) baseline similar to those that have been used in other text generative tasks [146, 37]. First, the baseline encodes the list of input triples using

<sup>2</sup>[https://meta.wikimedia.org/wiki/Wikidata/Notes/Language\\_fallback](https://meta.wikimedia.org/wiki/Wikidata/Notes/Language_fallback)



	Model	BLEU 1		BLEU 2		BLEU 3		BLEU 4		ROUGE <sub>L</sub>		METEOR	
		Valid.	Test	Valid.	Test	Valid.	Test	Valid.	Test	Valid.	Test	Valid.	Test
Arabic	KN	12.84	12.85	2.28	2.4	0.95	1.04	0.54	0.61	17.08	17.09	29.04	29.02
	KN <sub>ext</sub>	28.93	28.84	21.21	21.16	16.78	16.76	13.42	13.42	28.57	28.52	30.47	30.43
	IR	41.39	41.73	34.18	34.58	29.36	29.72	25.68	25.98	43.26	43.58	32.99	33.33
	IR <sub>ext</sub>	49.87	48.96	42.44	41.5	37.29	36.41	33.27	32.51	51.66	50.57	34.39	34.25
	Ours	53.61	54.26	47.38	48.05	42.65	43.32	38.52	39.20	64.27	64.64	45.89	45.99
	+ Copy	<b>54.10</b>	<b>54.40</b>	<b>47.96</b>	<b>48.27</b>	<b>43.27</b>	<b>43.60</b>	<b>39.17</b>	<b>39.51</b>	<b>64.60</b>	<b>64.69</b>	<b>46.09</b>	<b>46.17</b>
Esperanto	KN	18.12	17.8	6.91	6.64	4.18	4.0	2.9	2.79	37.48	36.9	31.05	30.74
	KN <sub>ext</sub>	25.17	24.93	16.44	16.3	11.99	11.92	8.77	8.79	44.93	44.77	33.77	33.71
	IR	43.01	42.61	33.67	33.46	28.16	28.07	24.35	24.3	46.75	45.92	20.71	20.46
	IR <sub>ext</sub>	<b>52.75</b>	<b>51.66</b>	43.57	42.53	37.53	36.54	33.35	32.41	58.15	57.62	31.21	31.04
	Ours	49.34	49.40	42.83	42.95	38.28	38.45	34.66	34.85	66.43	<b>67.02</b>	40.62	<b>41.13</b>
	+ Copy	50.22	49.81	<b>43.57</b>	<b>43.19</b>	<b>38.93</b>	<b>38.62</b>	<b>35.27</b>	<b>34.95</b>	<b>66.73</b>	66.61	<b>40.80</b>	40.74

Table 5.4: Automatic evaluation of our model against all other baselines using BLEU 1-4, ROUGE and METEOR for both Arabic and Esperanto Validation and Test set

TF-IDF followed by LSA [66]. For each item in the test set, we perform K-nearest neighbors to retrieve the vector from the training set that is the closest to this item and output its corresponding summary. Similar to KN baseline, we provide two versions of this baseline IR and IR<sub>ext</sub>.

### 5.1.5 Results and Discussion

We evaluate the generated summaries from our model and each of the baselines against their original counterparts from Wikipedia. Triples sets whose generated summaries are incomplete<sup>3</sup> (i.e. summaries for which the special end-of-summary <end> token is generated) are excluded from the evaluation. We use a set of evaluation metrics for text generation: BLEU [130], METEOR [31] and ROUGE<sub>L</sub> [98]. As displayed in Table 5.6, our model shows a significant enhancement compared to our baselines across the majority of the evaluation metrics in both languages. We achieve at least an enhancement of at least 5.25 and 1.31 BLEU 4 score in Arabic and Esperanto respectively over the IR<sub>ext</sub>, the strongest baseline. The introduction of the copy actions to our encoder-decoder architecture enhances our performance further by 0.61 – 1.10 BLEU (using BLEU 4). In general, our copy actions mechanism benefits the performance of all the competitive systems.

<sup>3</sup>Around  $\leq 1\%$  and  $2\%$  of the input validation and test triples sets in Arabic and Esperanto respectively led to the generation of summaries without the <end> token. We believe that this difference is explained by the limited size of the Esperanto dataset that increases the level of difficulty that the trained models (i.e. with or without Copy Actions) to generalize on unseen data.

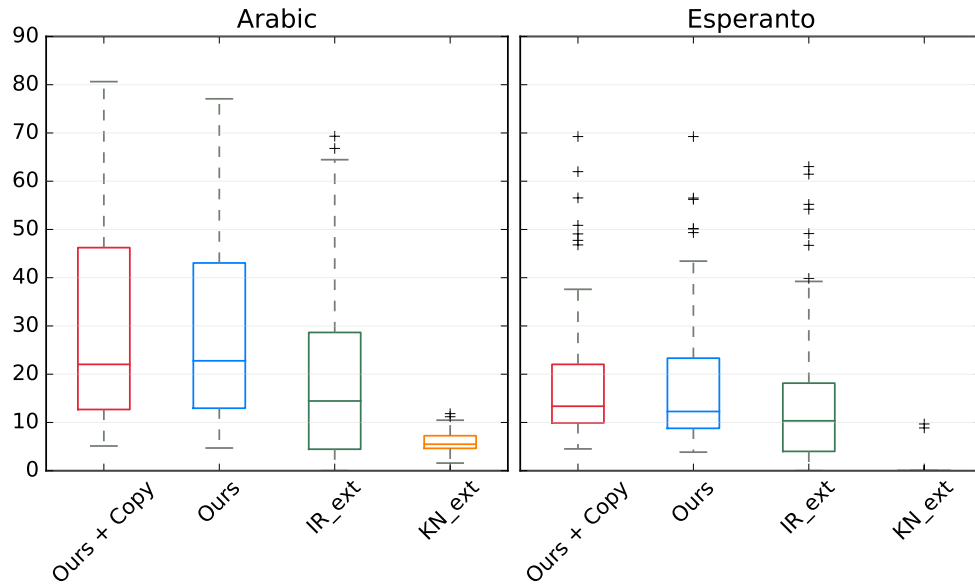


Figure 5.2: A box plot showing the distribution of BLEU 4 scores of all systems for each category of generated summaries.

**Generalisation Across Domains.** To investigate how well different models can generalize across multiple domains, we categorize each generated summary into one of 50 categories according to its main entity instance type (e.g. village, company, football player). We examined the distribution of BLEU-4 scores per category to measure how well the model generalizes across domains (Figure 5.2). We showed that i) the high performance of our system is not skewed towards some domains at the expense of others, and that ii) our model has a good generalization across domains – better than any other baseline. For instance, the over performance of  $IR_{ext}$  is limited to a few number of domains – plotted as the few outliers in Figure 5.2 for  $IR_{ext}$  –, despite its performance being much lower on average for all the domains.

Despite the fact that  $IR_{ext}$  achieves the highest recorded performance in a few domains (i.e.  $IR_{ext}$  outliers in Figure 5.2), its performance is much lower on average for all the domains.

The valuable generalisation of our model across domains is mainly due to the language model in the decoder layer of our model, which is more flexible than rigid templates and can adapt easier to multiple domains. Despite the fact that the Kneser-Ney

template-based baseline ( $KN_{\text{ext}}$ ) has exhibited competitive performance in a single-domain context [92], it is failing to generalize in our multi-domain text generation scenario. Unlike our approach,  $KN_{\text{ext}}$  does not incorporate the input triples directly for generating the output summary, but rather only uses them to replace the special tokens after a summary has been generated.

This might yield acceptable performance in a single domain, where most of the summaries share a very similar pattern.

However, it struggles to generate a different pattern for each input set of triples in multiple domain summary generation.

### 5.1.6 Conclusions

In this section, we showed that with the adaptation of the encoder-decoder neural network architecture for the generation of summaries we are able to overcome the challenges introduced by working with underserved languages. This is achieved by leveraging data from a structured knowledge base and careful data preparation in a multilingual fashion, which are of the utmost practical interest for our under-resourced task, that would have otherwise required a substantial additional amount of data. Our model was able to perform and generalize across domains better than a set of strong baselines of different nature including Language Modeling and Information Retrieval over templates. In the next section we integrate this study as a part of the Wikipedia ArticlePlaceholder project to generate introductory paragraphs for multilingual Wikipedias, this enables us to extend this study with an extensive crowd-sourcing study to qualitatively evaluate NLG from knowledge bases.

## 5.2 Mind the (Language) Gap: Generation of Multilingual Wikipedia Summaries from Wikidata for ArticlePlaceholders

The content of the most under-resourced Wikipedias is maintained by a limited number of editors – they cannot curate the same volume of articles as in the large Wikipedia communities. Part of this problem has been addressed by Wikidata, the KB supporting Wikipedia with structured data in a cross-lingual manner. Recently, Wikimedia introduced **ArticlePlaceholders** [81] in order to integrate Wikidata’s knowledge into the Wikipedias of underserved languages and help in reducing the language gap. ArticlePlaceholders display Wikidata triples in a tabular-based way in the target Wikipedia language and are currently deployed to 11 underserved Wikipedias<sup>4</sup>. When a user searches for a topic on Wikipedia that has a Wikidata item, but no Wikipedia article yet, they are led to the ArticlePlaceholder<sup>5</sup> on the topic. Compared to stub articles<sup>6</sup>, ArticlePlaceholders have the advantage of being dynamically updated in real time to accommodate information changes in Wikidata. This means less maintenance for small communities of editors. Since Wikidata is one central, language-independent place to edit information and each item or property has to be translated only once, any contribution in Wikidata has an impact on the ArticlePlaceholders. For example, an editor speaking only English can connect the existing items Q1299 (*The Beatles*) with the item Q145 (*United Kingdom*) via the property P495 (*country of origin*). This will automatically add the same triple with their Esperanto labels : *The Beatles – eldonit/ata en – Unuiĝinta Reĝlando*. Nonetheless, ArticlePlaceholders currently only display information in the form of tables.

In this section, we propose an automatic approach to enrich ArticlePlaceholders with textual summaries that can serve as a starting point for the Wikipedia editors to write their article. The summaries resemble the first sentence of a Wikipedia article, that gives a reader an overview of the topic. We adapt an end-to-end trainable model, which generates a monolingual textual summary (i.e. only in English) given a set of KB triples as input, for multilingual support. To this end, we introduce a new “property placeholders”

---

<sup>4</sup>cy, eo, lv, nn, ht, kn, nap, gu, or, sq, and bn

<sup>5</sup>Example as of online now, without the integration of generated summaries: <https://gu.wikipedia.org/wiki/special>AboutTopic/Q7186>

<sup>6</sup><https://en.wikipedia.org/wiki/Wikipedia:Stub>

feature and put them under distant supervision in order to enable our system to verbalise even rare or "unseen" entities. Since the summaries are generated explicitly based on the input triples, potential changes in the respective triples can manifest themselves immediately to the textual content of the summary without the inclusion of the translation loop. Furthermore, since we do not transfer any information from a source language, our model learns to generate Wikipedia content that captures the linguistic peculiarities of our target underserved Wikipedias.

We apply our model on two languages that have a severe lack of both editors and articles on Wikipedia: Esperanto and Arabic.

We propose a novel evaluation framework that assesses the usefulness of the summaries via a multitude of metrics, computed against strong baselines and involving readers and editors of underserved Wikipedias. We start our evaluation by measuring how close our synthesized summaries are to actual summaries in Wikipedia. We compare our model to two strong baselines of different natures: MT and a template-based solution. Our model substantially outperforms the baselines in all evaluation metrics in both Esperanto and Arabic. In addition, we developed three studies with the Wikipedia community, in which we ask for their feedback about the generated summaries, in terms of their fluency, appropriateness for Wikipedia, and engagement with editors. We believe that given the promising results achieved in the automatic and human evaluations, our approach along with the datasets, the baselines, and the experimental design of the human evaluation can serve as a starting point for the research community to further improve and assist in solving this critical task. Our code and experiments are available: <https://github.com/pvougliou/Mind-the-Language-Gap>.

### 5.2.1 Related Work

**Multilingual Text Generation** Many existing techniques for text generation and RDF verbalization rely on templates. These templates are generated using linguistic features such as grammatical rules [179], or are hand-crafted [54]. These approaches face many challenges when scaling for a language-independent system, as templates need to be fine-tuned to any new languages they are ported to. This is especially difficult for the few editors of underserved Wikipedias since templates need extra attention. They would have to create and maintain templates while this time could be invested in the creation of

an actual article. Recognizing this problem, the authors of [39, 44] introduce a distant-supervised approach to verbalize triples. The templates are learned from existing Wikipedia articles. This makes the approach more suitable for language-independent tasks. However, templates always assume that items will always have the appropriate triples to fill the slots of the template. This assumption is not always necessarily true. In our experiments, we implement a template-learning baseline and we show that adapting to the varying triples available can achieve better performance.

**Text Generation for Wikipedia** Sauper et al. and Pochmaply et al. proposed the generation of Wikipedia summaries by harvesting sentences from the Internet [149, 134]. Existing Wikipedia articles are used to automatically derive templates for the topic structure of the summaries and the templates are afterward filled using Web content. Such approaches are limited to only one or two domains and only in English. The lack of Web resources for underserved languages prevents these approaches to scale to undeserved languages in multiple domains [97]. Meanwhile, KBs have been used as a resource for NLG [22, 39, 118, 175]. These techniques leverage linguistic information from KBs to build a dataset of triples aligned with equivalent sentences from Wikipedia. This alignment is used at subsequent steps to train NLG systems.

The most relevant work to our proposed model are the recent approaches by Lebre et al. [92], Chisholm et al. [22], and Vougiouklis et al. [175], who all propose adaptations of the general encoder-decoder neural network framework [23, 165]. They use structured data from Wikidata and DBpedia as input and generate one sentence summaries that match the Wikipedia style in English in only a single domain. The first sentence of Wikipedia articles in a single domain exhibits a relatively narrow domain of language in comparison to other text generation tasks such as translation. However, Chisholm et al. [22] show that this task is still challenging and far from being solved. In contrast with these works, in our section we extend those research work to include open-domain, multilingual summaries.

**Evaluating Text Generation** Evaluating generated text is challenging and there have been different approaches proposed by the literature. Automatic scores [92], expert evaluation and crowdsourcing [90, 22] have been employed. Additionally, similar to Sauper et al. [149], we extend our evaluation to usefulness of the summaries for Wikipedia

editors by measuring the amount of reuse of the generated summaries. This concept has been widely investigated in fields such as journalism [26] and plagiarism detection [135].

### 5.3 Methods

We use a neural network in order to understand the impact of adding automatically generated text to ArticlePlaceholders in underserved language Wikipedias.

#### Our System

Our system is adapted from our encoder-decoder architecture introduced in [175] that has already been used on a similar text generative task. The architecture of the generative model is displayed in Figure 5.3. The encoder is a feed-forward architecture which encodes an input set of triples into a vector of fixed dimensionality. This is used at a later stage to initialise the decoder. The decoder is an RNN that uses Gated Recurrent Units (GRUs) [23] to generate the textual summary one token at a time.

An example is presented in Table 5.2. The ArticlePlaceholder provides our system with a set of triples about the Wikidata item of *Floridia* (i.e. Q490900 (Floridia) is either the subject or the object of the triples in the set). Figure 5.3 displays how our model generates a summary from those triples,  $f_1$ ,  $f_2$ , and  $f_3$ . A vector representation  $h_{f_1}$ ,  $h_{f_2}$ , and  $h_{f_3}$  for each of the input triples is computed by processing their subject, predicate and object. These vector representations are used to compute a vector representation for the whole input set  $h_{F_E}$ .  $h_{F_E}$ , along with the special start-of-summary `<start>` token, are used to initialise the decoder that sequentially predicts tokens (“[[Q490900, Floridia]]”, “estas”, “komunumo” etc.).

Formally, let  $F_E$  be the set of triples provided by the ArticlePlaceholder for the item  $E$  (i.e. item  $E$  is either the subject or the object of the triples in the set), our goal is to learn a model that generates a summary  $Y_E$  about  $E$ . We regard  $Y_E$  as a sequence of  $T$  tokens such that  $Y_E = y_1, y_2, \dots, y_T$  and compute the conditional probability  $p(Y_E|F_E)$ :

$$p(Y_E|F_E) = \prod_{t=1}^T p(y_t|y_1, \dots, y_{t-1}, F_E) . \quad (5.4)$$

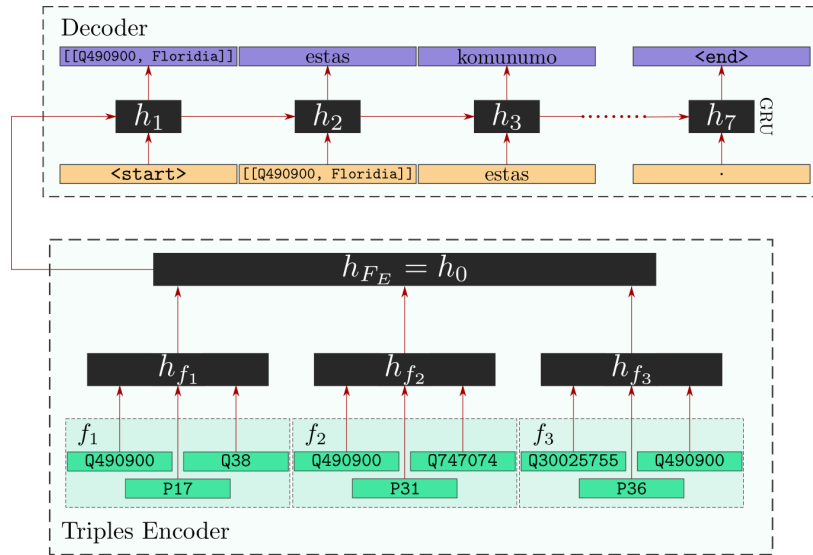


Figure 5.3: The triple encoder computes a vector representation for each one of the three input triples from the ArticlePlaceholder,  $h_{f_1}$ ,  $h_{f_2}$  and  $h_{f_3}$ . Subsequently, the decoder is initialized using the concatenation of the three vectors,  $[h_{f_1}; h_{f_2}; h_{f_3}]$ . The purple boxes represent the tokens of the generated summary. Each summary starts and ends with the respective start-of-summary  $\langle \text{start} \rangle$  and end-of-summary  $\langle \text{end} \rangle$  tokens.

### Generating a Summary

Our model learns to make a prediction about the next token by using the negative cross-entropy criterion. We define a maximum number of triples per summary. Input sets with fewer triples are padded with zero vectors, which are consistently ignored by the encoder. During training our architecture predicts the sequence of tokens that make up the summary. During testing, the ArticlePlaceholder provides our model with a set of unknown triples. After the vector representation  $h_{F_E}$  for the unknown set of triples is computed, we initialize the decoder with a special start-of-sequence  $\langle \text{start} \rangle$  token. We adopt a beam-search decoder [92, 165, 175] which provides us with  $B$ -most-probable summaries for each triple set  $F_E$ .

### Vocabulary Extensions

Each summary consist of words and mentions of named entities. Mapping those entities to words is hard since an entity can have several surface forms and the system may face rare/unseen entities at prediction time. We adopt the concept of *surface form tuples*



to learn a number of different verbalisations of the same entity in the summary [175]. In Table 5.2, `[[Q490900, Florida]]` in the vocabulary extended summary is an example of a surface form tuple where the entity `Q490900` is associated with the surface form of "Florida".

Additionally, we address the problem of learning embeddings for rare entities in text [102] by training our model to match the occurrence of rare entities in the text to the corresponding triple. To this end, we introduce *property placeholders*. The property placeholders are inspired by the *property-type placeholders* [175]. However, their applicability is much broader since they do not require any instance type-related information about the entities that appear in the triples. In the vocabulary extended summary of Table 5.2, `[[P17]]` is an example of property placeholder. In case it is generated by our model, it is replaced with the label of the object of the triple with which they share the same property (i.e. `Q490900 (Florida) P17 (stato) Q38 (Italia)`).

Further details regarding the fundamental components of our neural architecture, such as the triples encoder and the surface form tuples, can be found in the previous sections.

### 5.3.1 Training and Automatic Evaluation

Here, we describe the dataset that we built for our experiments along with the results of the automatic evaluation of our neural network architecture against the baselines.

**Dataset:** In order to train and evaluate our system, we created a new dataset for text generation from KB triples in a multilingual setting. We wish to explore the robustness of our approach to variable datasets with respect to language complexity and size of available training data. Consequently, we worked with two linguistically distinct Wikipedias of different sizes and different language support in Wikidata [84].

This dataset aligns Wikidata triples with the first, introductory sentence of its corresponding Wikipedia articles. For each Wikipedia article, we extracted and tokenized the first sentence using a multilingual Regex tokenizer from the NLTK toolkit [12]. Afterwards, we retrieved the corresponding Wikidata item to the article and queried all triples where the item appeared as a subject or an object in the Wikidata truthy dump<sup>7</sup>.

---

<sup>7</sup><https://dumps.wikimedia.org/wikidatawiki/entities/>

In order to create the *surface form tuples* (i.e. sub section 5.3), we identify occurrences of entities in the text along with their verbalisations. We rely on keyword matching against labels from Wikidata from the corresponding language, due to the lack of reliable entity linking tools for underserved languages.

For the *property placeholders* (described in more detail in sub section 5.3), we use the distant-supervision assumption for relation extraction [114]. After identifying the rare entities that participate in relations with the main entity of the article, they are replaced from the introductory sentence with their corresponding property placeholder tag (e.g. [[P17]] in Table 5.2). During testing, any property placeholder token that is generated by our system is replaced by the label of the entity of the relevant triple (i.e. triple with the same property as the generated token).

**Automatic Evaluation:** To evaluate how well our system generates textual summaries for Wikipedia, we evaluated the generated summaries against two baselines on their original counterparts from Wikipedia. We use a set of evaluation metrics for text generation BLEU 1, BLEU 2, BLEU 3, BLEU 4, METEOR and ROUGE<sub>L</sub>. BLEU calculates n-gram precision multiplied by a brevity penalty which penalizes short sentences to account for word recall. METEOR is based on the combination of uni-gram precision and recall, with recall weighted over precision. It extends BLEU by including stemming, synonyms and paraphrasing. ROUGE<sub>L</sub> is a recall-based metric which calculates the length of the most common subsequence between the generated summary and the reference.

### Baselines for Automatic Evaluation

Due to the variety of approaches for text generation, we demonstrate the effectiveness of our system by comparing it against two baselines of different nature.

**Machine Translation (MT)** For the MT baseline, we used Google Translate on English Wikipedia summaries. Those translations are compared to the actual target language’s Wikipedia entry. This limits us to articles that exist in both English and the target language. In our dataset, the concepts in Esperanto and Arabic that are not covered by English Wikipedia account for 4.3% and 30.5% respectively. This indicates the content coverage gap between different Wikipedia languages [71].

		#P	#S	#P: S>50%	Avg #S/P	All Ann.
Arabic	Fluency	27	60	5	15.03	406
	Approp.	27	60	5	14.78	399
	Editors	7	30	2	4	33
Esperanto	Fluency	27	60	3	8.7	235
	Approp.	27	60	3	8.63	233
	Editors	8	30	2	4.75	38

Table 5.5: Participation Numbers: Total number of Participants ( $P$ ), Total number of Sentences ( $S$ ), Number of  $P$  that evaluated at least 50% of  $S$ , and average number of  $S$  evaluated per  $P$

**Template Retrieval (TP)** Similar to template-based approaches for text generation [146, 44], we build a template-based baseline that retrieves an output summary from the training data based on the input triples. First, the baseline encodes the list of input triples that corresponds to each summary in the training/test sets into a sparse vector of TF-IDF weights [78]. Afterwards, it performs LSA [66] to reduce the dimensionality of that vector. Finally, for each item in the test set, we employ the K-nearest neighbors algorithm to retrieve the vector from the training set that is the closest to this item. The summary that corresponds to the retrieved vector is used as the output summary for this item in the test set. We provide two versions of this baseline. The first one (TP) retrieves the raw summaries from the training dataset. The second one (TP<sub>ext</sub>) retrieves summaries with the special tokens for vocabulary extension. A summary can act as a template after replacing its entities with their corresponding *Property Placeholders* (see Table 5.2).

### 5.3.2 Community Study

Automatic measures of text quality such as BLEU can give an indication of how close a generated text is to the source of a summary. Complementary, working with humans is generally more trusted when it comes to quality evaluation of generated text, and captures the direct response of the community. We ran a community study for a total of 15 days. To address the question whether the textual summaries can match the quality of Wikipedia, we define text quality as fluency and appropriateness. Fluency describes the quality in terms of understandability and grammatical correctness. Appropriateness describes how well a summary fits into Wikipedia, i.e. whether a reader can identify it as part of a Wikipedia article. We assess editors reuse to answer whether we can generate

summaries that are useful for Wikipedia editors. Our evaluation targets two different communities: (1) *readers*: Any speaker of Arabic and Esperanto, that reads Wikipedia, independent of their activity on Wikipedia, and (2) *editors*: any active contributor to Arabic and Esperanto Wikipedia. Readers were asked to fill one survey combining fluency and appropriateness. Editors were also asked to fill an additional survey<sup>8</sup>. To sample only participants with previous activity on Wikipedia, we asked them for their reading and editing activity on Wikipedia. The survey instructions<sup>9</sup> and announcements<sup>10</sup> were translated in Arabic and Esperanto.

**Recruitment** For the recruitment of readers, we wanted to reach fluent speakers of the language. For Arabic, we got in contact with Arabic speaking researchers from research groups working on Wikipedia related topics. For Esperanto, as there are fewer speakers and they are harder to reach, we promoted the survey on social media such as Twitter and Reddit<sup>11</sup> using the researchers’ accounts. For the recruitment of editors, we posted on the editors’ mailing-lists<sup>12</sup>. Additionally, for Esperanto we posted on the Wikipedia discussion page<sup>13</sup>. The Arabic editors survey was also promoted at WikiArabia, the conference for the Arabic speaking Wikipedia community. The numbers of participation in all surveys can be found in Table 5.5.

**Fluency** We answer whether we can generate summaries that match the quality and style of Wikipedia content in a study with 54 Wikipedia readers from two different Wikipedia languages. We created a corpus consisting of 60 summaries of which 30 are generated through our approach, 15 are from news, 15 from Wikipedia summaries of the training dataset. For news in Esperanto, we chose introduction sentences of articles in the Esperanto version of *Le Monde Diplomatique*<sup>14</sup>. For news in Arabic, we chose introduction sentences of the RSS feed of BBC Arabic<sup>15</sup>. Each participant was asked

<sup>8</sup>Example questions: <https://github.com/pvougliou/Mind-the-Language-Gap/tree/master/crowdevaluation/Examples>

<sup>9</sup>All instructions for the surveys: <https://tinyurl.com/y7cgmesk>

<sup>10</sup><https://github.com/luciekaffee/Announcements>

<sup>11</sup>[https://www.reddit.com/r/Esperanto/comments/75rytb/help\\_in\\_a\\_study\\_using\\_ai\\_to\\_create\\_esperanto/](https://www.reddit.com/r/Esperanto/comments/75rytb/help_in_a_study_using_ai_to_create_esperanto/)

<sup>12</sup>Esperanto: [eliso@lists.wikimedia.org](mailto:eliso@lists.wikimedia.org), Arabic: [wikiar-l@lists.wikimedia.org](mailto:wikiar-l@lists.wikimedia.org)

<sup>13</sup>[https://eo.wikipedia.org/wiki/Vikipedio:Diskutejo/Diversejo#Help\\_in\\_a\\_study\\_improving\\_Esperanto\\_text\\_for\\_Editors](https://eo.wikipedia.org/wiki/Vikipedio:Diskutejo/Diversejo#Help_in_a_study_improving_Esperanto_text_for_Editors)

<sup>14</sup><http://eo.monediplo.com/>, accessed 28. September 2017

<sup>15</sup><http://feeds.bbci.co.uk/arabic/middleeast/rss.xml>, accessed 28 Sep 2017

Model	BLEU-1		BLEU-2		BLEU-3		BLEU-4		ROUGE <sub>L</sub>		METEOR	
	valid	test	valid	test	valid	test	valid	test	valid	test	valid	test
KN	12.84	12.85	2.28	2.4	0.95	1.04	0.54	0.61	17.08	17.09	29.04	29.02
KN <sub>ext</sub>	28.93	28.84	21.21	21.16	16.78	16.76	13.42	13.42	28.57	28.52	30.47	30.43
MT	31.12	33.48	19.31	21.12	12.69	13.89	8.49	9.11	29.96	30.51	31.05	30.1
Ar IR	41.39	41.73	34.18	34.58	29.36	29.72	25.68	25.98	43.26	43.58	32.99	33.33
IR <sub>ext</sub>	49.87	48.96	42.44	41.5	37.29	36.41	33.27	32.51	51.66	50.57	34.39	34.25
Our Model	<b>53.18</b>	<b>52.94</b>	<b>45.86</b>	<b>45.64</b>	<b>40.38</b>	<b>40.21</b>	<b>35.7</b>	<b>35.55</b>	<b>57.9</b>	<b>57.99</b>	<b>39.22</b>	<b>39.37</b>
KN	18.12	17.8	6.91	6.64	4.18	4.0	2.9	2.79	37.48	36.9	31.05	30.74
KN <sub>ext</sub>	25.17	24.93	16.44	16.3	11.99	11.92	8.77	8.79	44.93	44.77	33.77	33.71
MT	5.35	5.47	1.62	1.62	0.59	0.56	0.26	0.23	4.67	4.79	0.66	0.68
Es IR	43.01	42.61	33.67	33.46	28.16	28.07	24.35	24.3	46.75	45.92	20.71	20.46
IR <sub>ext</sub>	52.75	51.66	43.57	42.53	37.53	36.54	33.35	32.41	58.15	57.62	<b>31.21</b>	<b>31.04</b>
Our Model	<b>56.51</b>	<b>56.96</b>	<b>47.72</b>	<b>48.1</b>	<b>41.8</b>	<b>42.13</b>	<b>37.24</b>	<b>37.52</b>	<b>64.36</b>	<b>64.69</b>	28.35	28.76

Table 5.6: Automatic evaluation of our model against all other baselines using BLEU1-4, ROUGE and METEOR for both Arabic and Esperanto Validation and Test set

to assess the fluency of the text. We employ a scale from 0 to 6, where: **(6) Excellent:** the given sentence has no grammatical flaws and the content can be understood with ease; **(3) Moderate:** the given sentence is understandable, but has minor grammatical issues; **(0) Non-understandable:** the given sentence cannot be understood. For each sentence, we calculate the mean quality given by all participants and then averaging over all summaries in each corpus.

**Appropriateness** As we used the same survey for both fluency and appropriateness, participants answered questions regarding the appropriateness over the same set of sentences. They were asked to assess whether the displayed sentence could be part of a Wikipedia article. We test whether a reader can tell the difference from just one sentence whether a text is appropriate for Wikipedia, using the news sentences as a baseline. This gives us an insight on whether the text produced by the neural network “feels” like Wikipedia text (appropriateness). Participants were asked not to use any external tools for this task. Readers have just two options to choose from (Yes and No).

**Editors Reuse** We randomly choose 30 items from our test set. For each item, each editor was offered the generated summary and its corresponding set of triples and was asked to write a paragraph of 2 or 3 sentences. Editors had the freedom to copy from the generated summary, or completely work from scratch. We assessed how editors used our generated summaries in their work by measuring the amount of text reuse. To

quantify the amount of reuse in text we use the Greedy String-Tiling (GST) algorithm [183]. GST is a substring matching algorithm that computes the degree of reuse or copy from a source text and a dependent one. GST is able to deal with cases when a whole block is transposed, unlike other algorithms such as the Levenshtein distance, which calculates it as a sequence of single insertions or deletions rather than a single block move. Given a generated summary  $S = s_1, s_2, \dots$  and an edited one  $D = d_1, d_2, \dots$ , each consisting of a sequence of tokens, GST will identify a set of disjoint longest sequences of tokens in the edited text that exist in the source text (called *tiles*)  $T = \{t_1, t_2, \dots\}$ . It is expected that there will be common stop words appearing in both the source and the edited text. However, we are rather interested in knowing how much of real structure of the generated summary is being copied. Thus, we set minimum match length factor  $mml = 3$  when calculating the tiles, s.t.  $\forall t_i \in T : t_i \subseteq S \wedge t_i \subseteq D \wedge |t_i| \geq mml$  and  $\forall t_i, t_j \in T | i \neq j : t_i \cap t_j = \emptyset$ . This means that copied sequences of single or double words will not count in the calculation of reuse. We calculate a reuse score  $gstscore$  by counting the lengths of the detected tiles, and normalize by the length of the generated summary.

$$gstscore(S, D) = \frac{\sum_{t_i \in T} |t_i|}{|S|} \quad (5.5)$$

We classify each of the edits into three groups according to the  $gstscore$  as proposed by [26]: 1) **Wholly Derived (WD)**: the summary structure has been fully reused in the composition of the editor’s text ( $gstscore \geq 0.66$ ); 2) **Partially Derived (PD)**: the summary has been partially used ( $0.66 > gstscore \geq 0.33$ ); 3) **Non Derived (ND)**: The summary has been changed completely ( $0.33 > gstscore$ ).

### 5.3.3 Results and Discussions

#### Automatic Evaluation

As displayed in Table 5.6, our model shows a significant enhancement compared to our baselines across the majority of the evaluation metrics in both languages. We achieve a **3.01** and **5.11** enhancement in BLEU 4 score in Arabic and Esperanto respectively over  $TP_{ext}$ , the strongest baseline. MT of English summaries is not competitive. We attribute this result to the differences in the way of writing across different Wikipedia languages – this inhibits MT from being sufficient for Wikipedia document generation. The results show that generating language directly from the knowledge base triples is a much more







		Fluency		Appropriateness	
		Mean	SD	Part of Wikipedia	
Arabic	Our model	4.7	1.2	77%	
	Wikipedia	4.6	0.9	74%	
	News	5.3	0.4	35%	
Esperanto	Our model	4.5	1.5	69%	
	Wikipedia	4.9	1.2	84%	
	News	4.2	1.2	52%	

Table 5.7: Results for fluency and appropriateness

suitable approach.

### 5.3.4 Community Study

We present the results of the community study in order to find whether we could generate textual summaries that match the quality and style of Wikipedia and can support editors.

**Fluency (Table 5.7)** Overall, the quality of our generated summaries is high (4.7 points in average in Arabic, 4.5 in Esperanto). In Arabic, 63.3% of the summaries were evaluated to have at least 5 (out of 6) in average. In Esperanto, 50% of the summaries have at least a quality of 5 (out of 6) in average, with 33% of all summaries given a score of 6 by all participants. This means the majority of our summaries is highly understandable and grammatically correct. Furthermore, our generated summaries are also considered by participants to have a similar average quality as Wikipedia summaries and news from widely read media organizations.

**Appropriateness (Table 5.7)** 77% (resp. 69%) of the generated Arabic (resp. Esperanto) summaries were categorized as being part of Wikipedia. In comparison, news sentences were identified more likely to not fit. In only 35% (Arabic) and 52% (Esperanto) of cases, readers have mistaken them for Wikipedia sentences. Wikipedia sentences were clearly recognized as such (77% and 84%) with scores that are closely matching the one from the generated summaries from our model. Wikipedia has a certain writing style, that seems to differ clearly from news. Our summaries are able to

	Category	Examples	Percentage
Arabic	WD	<p>خماسي كلوريد الزرنيخ مركب كيميائي له الصيغة (كلمة ناقصة )، ويكون على شكل بلورات بيضاء <sup>A</sup></p> <p>خماسي كلوريد الزرنيخ هو مركب كيميائي له الصيغة (AlCl<sub>3</sub>2085)، ويكون على شكل بلورات بيضاء. <sup>B</sup></p>	45.45%
	PD	<p>بيتش باتوم (أومايو) بالإنجليزية (كلمة ناقصة) Ohio هي منطقة سكنية تقع في الولايات المتحدة في (كلمة ناقصة).</p> <p>بيتش باتوم (بالإنجليزية: Beach Batom) هي قرية تقع في الولايات المتحدة الأمريكية في بروك كاونتي.</p>	33.33%
	ND	<p>دير علا هي بلدة تقع في جنوب غرب إيران.</p> <p>دير علا، أو بيشر، هي قرية أردنية</p>	21.21%
Esperanto	WD	<p>Zederik estas komunumo en la nederlanda provinco Zuid-Holland <sup>b</sup></p> <p>Zederik estas komunumo en la nederlanda provinco Zuid-Hooland kaj estas ĉirkaŭata de la municipoj Lopik kaj Zederik.</p>	78.98%
	PD	<p>Nova Pádua estas municipo en la brazila subŝtato Suda Rio-Grando, kiu havis (manka nombro) loĝantojn en (jaro).</p> <p>Nova Pádua estas municipo en la brazila subŝtato Suda Rio-Grando. <sup>i</sup></p>	15.79%
	ND	<p>Ibiúna estas municipo de la brazila subŝtato San-Paŭlio, kiu taksis (manka nombro) enloĝantojn en (jaro).</p> <p>Ibiúna estas brazila [[municipo]] kiu troviĝas en la administra unuo [[San-Paŭlio]].</p>	5.26%

Table 5.8: Percentage of summaries in each category of reuse in Arabic and Esperanto. An example is provided for each category containing a generated summary (top) and after it was edited (bottom). Solid lines represent reused tiles, while dashed lines represent overlapping sub-sequences smaller than *mml* and not contributing to the *gstscores* calculation.

reflect this writing style, being more likely evaluated as Wikipedia sentences than the news baseline – we can expect the generated summaries to melt seamlessly with other Wikipedia content.

**Editors Reuse (Table 5.8)** Our summaries were highly reused. **79%** of the Arabic generated summaries and **93%** of the Esperanto generated summaries were either wholly (**WD**) or partially (**PD**) reused by editors. For the wholly derived edits, editors tended to copy the generated summary with minimal modifications such as Table 5.8 subsequences A and B in Arabic or subsequence G in Esperanto. One of the common things that hampers the full reusability are "rare" tokens, in Arabic and (*mankas vorto*) in Esperanto. Usually, these tokens are yielded when the output word is not in the model vocabulary, it has not been seen frequently by our model such as names in different languages. As it can be seen in tiles *E* and *D* in the Arabic examples in Table 5.8, editors prefer in those cases to adapt the generated sentences. This can also go as far as making the editor to delete the whole subsentence if it contains a high number of such tokens (subsequence *H* in Table 5.8). By examining our generated summaries we find that such missing tokens are more likely to appear in Arabic than in Esperanto (2.2



times more). The observed reusability by editors of the Esperanto generated summaries (78.98% **WD**) in comparison to Arabic (45.45% **WD**) can be attributed to this. This can be explained as follows. First, the significant larger vocabulary size of Arabic, which lowers the probability of a word to be seen by the Arabic model. Second, since the majority of rare tokens are named entities mentioned in foreign languages and since the Latin script of Esperanto is similar to many other languages, the Esperanto model has an advantage over the Arabic one when capturing words representing named entities.

### **5.3.5 Conclusion**

We introduce a system that extends Wikipedia’s ArticlePlaceholder with multilingual summaries automatically generated from Wikidata triples for underserved language on Wikipedia. We show that with the encoder-decoder architecture that we propose is able to perform better than strong baselines of different natures, including MT and a template-based baseline. We ran a community evaluation study to measure to what extent our summaries match the quality and style of Wikipedia articles, and whether they are useful in terms of reuse by Wikipedia editors. We show that members of the targeted language communities rank our text close to the expected quality standards of Wikipedia, and are likely to consider the generated text as part of Wikipedia. Lastly, we found that the editors are likely to reuse a large portion of the generated summaries, thus emphasizing the usefulness of our approach to its intended audience.

## Chapter 6

# Conclusions and Future work

### 6.1 Summary and Conclusions

In the previous sections I demonstrated several contributions to tackle the three main challenges discussed in chapter 1 that prohibit the extensibility of Question Answering systems. These contributions have shown potential to increase the robustness of question answering systems to overcome the rapid dynamicity and evolution of web data. In real life scenarios, question answering systems in production get challenged daily by new entities and relation types to be represented, current strategies of building question answering systems do not have the capacity to handle this by themselves. Thus, the current solution for this is to support by a continuous engineering process that manually inserts new predicates and entity types to the background ontology of that question answering system. This engineering process is tedious and expensive and always requires domain experts. In chapter 3 we have demonstrated two solutions that facilitate solving this problem within the task of Relation Discovery.

Our first contribution in chapter 3 is a new relation discovery system that enables to better find, represent and cluster relation mentions in an supervised way in a large unstructured corpus. Our proposed relation discovery system is equipped with two novel components: the re-weighting of word embeddings based on the dependency parse tree of the sentence and an individual feature reduction of each feature before passing them to the clustering phase. Our method surpasses the state-of-the-art for relation clustering

by 5.8% pairwise F1 score.

Second, to overcome the very low recall issue of relation discovery methods that rely on sentences with a mention of two entities, we have developed a high recall approach for predicate extraction which enables covering up to 16 times more sentences in a large corpus. Our proposed method achieves 28% improvement over the highest recall OpenIE system without any compromises with respect to precision. This means that our approach is capable of extracting larger number of relation mentions in a corpus than traditional relation discovery methods.

In chapter 4 we tackled the problem of lack of training and evaluation datasets. In the first section we introduced one of the major contributions of this thesis which is *T-REx* an automatically built dataset for alignments of sentences in free text and structured knowledge base relations from Wikidata. *T-REx* is two orders of magnitude larger than the largest available alignments to the community and covers 2.5 times more predicates. These efforts behind were made to overcome the challenges we faced after working on the problem of relation discovery in chapter 3 – we noticed that most of the evaluation datasets aligning text and knowledge base triples were either limited in size or coverage or suffering of low quality. Alongside with the dataset we release also an extensive evaluation of the T-REx dataset and the framework equipped with a modular architecture that enables building similar datasets for different applications while replacing some of the components inside (e.g. domain specific entity linkers). During the course of this thesis, T-REx was utilized several times for several applications such as building datasets for providing textual mentions for Zeroshot Question Generation, as shown in section 2 chapter 4, or Multilingual Summary Generation from structured knowledge bases as in chapter 5. For the latter, we extended the dataset to include alignments for Arabic and Esperanto languages beside English.

In the second part of chapter 4 we tackled the problem of lack of training data for Question Answering systems over knowledge bases. Data augmentation through question generation from knowledge graphs has been studied before [152] to increase the size of training datasets. Although those techniques have proven to enhance the performance of question answering systems, these techniques do not make question answering systems to answer questions about relation and entities types beyond the existing ones in their initial training datasets. Thus, in this contribution, we study the viability of generating questions in a zeroshot setup, where we proposed a system that is capable of

generating questions for relation and entity types that remain unseen during test time. Those generated questions alongside with their input triples can be used to enable question answering systems to answer questions beyond the set of relations and entity types that exist in their original training data. To this end, we proposed an encoder-decoder architecture, paired with an original part-of-speech copy action mechanism to generate questions. The input of this model is a knowledge base triple embedded with a knowledge base embeddings technique, paired with 3 statements mentioning each part of this input triple in a separate statement; we refer to them as "textual mentions". Those textual mentions have proven to aid the encoder-decoder model to find the suitable lexicalization to express those unseen relations and entity types, specially when they were never mentioned before during training time. We relied on the distance supervision technique in the T-REx framework (chapter 4) to harvest those textual mentions in an unsupervised way. Through a set of experiments in a zero-shot setup, our proposed model showed to outperform a set of strong baselines indicating that this methodology can be very useful for expanding current question answering systems to answer questions beyond those in their training datasets.

In chapter 5, we looked into the problem of generating textual summaries to describe knowledge base entities. Current question answering systems display answers and extra information for a specific entity by extracting summaries written by humans from the web (e.g. from Wikipedia). These summaries are not always available for all entities especially for language with limited content on the web, thus a model that can automatically generate summaries for knowledge base entities can overcome this problem. We explored the viability of automatically generating multilingual textual summaries for entities in knowledge bases. As an extreme case to manifest the lack of training data we chose Arabic and Esperanto for our study as an example of underserved languages on the web with limited number of Wikipedia articles available. We proposed an encoder-decoder architecture equipped with novel copy actions that performs on entity and predicate labels in the input knowledge base. Our proposed architecture takes a set of Knowledge base triples describing the target entity as an input, and outputs a textual summary for this entity one word each time in the target language. To train this model we bootstrap a dataset by extending the T-REx framework 4 to extract alignments between knowledge base triples from Wikidata and sentences from the Arabic and the Esperanto Wikipedias. Our proposed model outperforms a set of strong baselines for language

generation from structured data, over a set of well established metrics for NLG evaluation.

In the second part of chapter 5 we perform an extensive study with active Wikipedia editors from the Arabic and the Esperanto Wikipedia communities to evaluate the correctness and usefulness of the generated summaries. For this we used the metric Greedy String Tiling (GST) to measure the usefulness of the generated summaries in terms of the percentage of reuse when writing a full introductory paragraph about the same entity.

## 6.2 Future Directions

In this section I discuss the limitations of our techniques, topics beyond this thesis, and directions to build upon this work in the future.

### 6.2.1 Data Augmentation Using Self-training

As discussed in chapter 1 and chapter 4 one of the defacto ways to automatically generate training datasets for relation extraction is the distant supervision assumption. This assumption trades lower accuracy and coverage with respect to manual annotations, but with the benefits of being entirely automatic.

The lack of training data is a well studied problem in Machine Learning and several solutions could be promising to tackle this problem within the framework of relation extraction, given the existence of a small start seed of annotated examples. Self-training [108] is one proposed techniques in which predictions of training model on a small set of annotated examples is used to pseudo annotate unlabeled examples (which are in theory easier to obtain), and then the most confidence annotations are fed back as additional training data for the model. This process is repeated to maximize the model predictions on a separate validation set. Since confidence scores of modern neural networks are known to be not calibrated [64], modern self-training techniques nowadays rely on an ensemble architectures to yield unbiased accurate predictions; this has led to the emergence of other techniques such as co-training and tri-training [201], tri-training with disagreement [157] and multi-task tri-training [143]. Those techniques have shown a large success for NLP tasks such as document classification and parsing specially under a domain shift. However they have not been yet explored for tasks such as relation extraction.

### **6.2.2 Data Augmentation for NLG Through Back Translation**

For generation, several techniques from machine translation can be utilized for providing data augmentation, one of those techniques are Back Translation [151, 42, 91] which has proven to be effective especially for low-resource languages. Back translation augments the parallel corpora with sentences back translated automatically from a monolingual corpora in the target language using a separate back translator model. While back translation in general is modeled to work on machine translation tasks where the input and the output to the model are sequences of tokens, it might be worth investigating if this methodology can be adapted for natural language generation from structured data, for example using a relation extractor instead of the back-translator.

### **6.2.3 Handling Domain Shift**

Most of the use cases we chose in the course of this thesis were applied to the general domain of knowledge (encyclopedia domain). This however might not be the case in real life applications where question answering systems might shift into a more specific domain such as medicine or legal documents. This might introduce the problem of domain shift. Machine learning models are known to suffer significant performance loss when exposed to domain foreign examples [86, 13], even without the introduction of new label classes, which is the problem we tackled in several application in this thesis. Luckily domain adaptation is a well studied problem in machine learning literature. There are several solutions that are proposed such as domain adversarial adaptation [55, 169], or using previously mentioned self-training and self-ensembling techniques [147, 143, 52].

# Bibliography

- [1] Representing Text for Joint Embedding of Text and Knowledge Bases. volume 15.
- [2] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [3] Gabor Angeli, Melvin Johnson Premkumar, and Christopher D Manning. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL*, pages 26–31, 2015.
- [4] Isabelle Augenstein, Diana Maynard, and Fabio Ciravegna. Distantly supervised web relation extraction for knowledge base population. *Semantic Web*, 7(4):335–349, 2016.
- [5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

- [6] Michele Banko, Michael J Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. Open Information Extraction from the Web. In *The twentieth international joint conference on artificial intelligence, IJCAI 2007*, pages 2670–2676, Hyderabad, India, 2007.
- [7] Michele Banko and Oren Etzioni. The tradeoffs between open and traditional relation extraction. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pages 28–36, 2008.
- [8] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 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, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 238–247, 2014.
- [9] Anja Belz and Ehud Reiter. Comparing automatic and human evaluation of NLG systems. In *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy, 2006*.
- [10] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1171–1179, 2015.
- [11] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, pages 932–938, 2000.
- [12] Steven Bird. NLTK: the natural language toolkit. In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006, 2006*.



- [13] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic, 2007*.
- [14] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.
- [15] Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. Large-scale simple question answering with memory networks. *CoRR*, abs/1506.02075, 2015.
- [16] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.
- [17] Gerlof Bouma. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, pages 31–40, 2009.
- [18] Martin Brümmer, Milan Dojchinovski, and Sebastian Hellmann. Dbpedia abstracts: A large-scale, open, multilingual nlp training corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France.
- [19] Angel X. Chang and Christopher Manning. Sutime: A library for recognizing and normalizing time expressions. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012.
- [20] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. *CoRR*, abs/1704.00051, 2017.
- [21] Danqi Chen and Christopher D. Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 740–750, 2014.

- [22] Andrew Chisholm, Will Radford, and Ben Hachey. Learning to generate one-sentence biographies from wikidata. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 633–642, Valencia, Spain, April 2017. Association for Computational Linguistics.
- [23] Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.
- [24] Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. An analysis of open information extraction based on semantic role labeling. In *Proceedings of the 6th International Conference on Knowledge Capture (K-CAP 2011), June 26-29, 2011, Banff, Alberta, Canada*, pages 113–120, 2011.
- [25] Janara Christensen, Stephen Soderland Mausam, Stephen Soderland, and Oren Etzioni. Towards coherent multi-document summarization. In *HLT-NAACL*, pages 1163–1173. Citeseer, 2013.
- [26] Paul D. Clough, Robert J. Gaizauskas, Scott S. L. Piao, and Yorick Wilks. ME-TER: MEasuring TExt Reuse. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.*, pages 152–159, 2002.
- [27] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.
- [28] Luciano Del Corro and Rainer Gemulla. Clausie: clause-based open information extraction. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, pages 355–366, 2013.
- [29] Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC 2006, Genoa, Italy, May 22-28, 2006.*, pages 449–454, 2006.

- [30] Thomas Demeester, Tim Rocktäschel, and Sebastian Riedel. Lifted rule injection for relation embeddings. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1389–1399, 2016.
- [31] Michael J. Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT@ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA*, pages 376–380, 2014.
- [32] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1811–1818, 2018.
- [33] Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 875–886, 2017.
- [34] Xin Dong, Evgeniy Gabrilovich, Jeremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 601–610, 2014.
- [35] Cícero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. Classifying relations by ranking with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 626–634, 2015.
- [36] Xinya Du, Junru Shao, and Claire Cardie. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting*

- of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1342–1352, 2017.
- [37] Xinya Du, Junru Shao, and Claire Cardie. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1342–1352, 2017.
- [38] Daniel Duma and Ewan Klein. Generating natural language from linked data: Unsupervised template extraction. In *Proceedings of the 10th International Conference on Computational Semantics, IWCS 2013, March 19-22, 2013, University of Potsdam, Potsdam, Germany*, pages 83–94, 2013.
- [39] Daniel Duma and Ewan Klein. Generating Natural Language from Linked Data: Unsupervised template extraction. In *IWCS*, pages 83–94, 2013.
- [40] Susan T Dumais. Latent semantic analysis. *Annual review of information science and technology*, 38(1):188–230, 2004.
- [41] Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 334–343, 2015.
- [42] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 489–500, 2018.
- [43] Basil Ell and Andreas Harth. A language-independent method for the extraction of RDF verbalization templates. In *INLG 2014 - Proceedings of the Eighth International Natural Language Generation Conference, Including Proceedings of the INLG and SIGDIAL 2014 Joint Session, 19-21 June 2014, Philadelphia, PA, USA*, pages 26–34, 2014.

- [44] Basil Ell and Andreas Harth. A language-independent method for the extraction of RDF verbalization templates. In *INLG 2014 - Proceedings of the Eighth International Natural Language Generation Conference, Including Proceedings of the INLG and SIGDIAL 2014 Joint Session, 19-21 June 2014, Philadelphia, PA, USA*, pages 26–34, 2014.
- [45] Hady ElSahar, Elena Demidova, Simon Gottschalk, Christophe Gravier, and Frédérique Laforest. Unsupervised open relation extraction. In *The Semantic Web: ESWC 2017 Satellite Events - ESWC 2017 Satellite Events, Portorož, Slovenia, May 28 - June 1, 2017, Revised Selected Papers*, pages 12–16, 2017.
- [46] Hady ElSahar, Christophe Gravier, and Frédérique Laforest. Zero-shot question generation from knowledge graphs for unseen predicates and entity types. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 218–228, 2018.
- [47] Hady ElSahar, Pavlos Vougiouklis, Arslan Remaci, Christophe Gravier, Jonathon S. Hare, Frédérique Laforest, and Elena Simperl. T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018.*, 2018.
- [48] Martha Evens and Joel Michael. One-on-one tutoring by humans and machines. *Computer Science Department, Illinois Institute of Technology*, 2006.
- [49] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics, 2011.
- [50] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1535–1545, 2011.

- [51] Anthony Fader, Luke S Zettlemoyer, and Oren Etzioni. Paraphrase-Driven Learning for Open Question Answering. In *ACL (1)*, pages 1608–1618. Citeseer, 2013.
- [52] Geoffrey French, Michal Mackiewicz, and Mark H. Fisher. Self-ensembling for visual domain adaptation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.
- [53] Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. FACC1: Freebase annotation of ClueWeb corpora, Version 1, June 2013.
- [54] Dimitrios Galanis and Ion Androutsopoulos. Generating Multilingual Descriptions from Linguistically Annotated OWL Ontologies: The NaturalOWL system. In *Proceedings of the Eleventh European Workshop on Natural Language Generation*, pages 143–146. Association for Computational Linguistics, 2007.
- [55] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17:59:1–59:35, 2016.
- [56] Dimitra Gkatzia and Saad Mahamood. A snapshot of NLG evaluation practices 2005 - 2014. In *ENLG 2015 - Proceedings of the 15th European Workshop on Natural Language Generation, 10-11 September 2015, University of Brighton, Brighton, UK*, pages 57–60, 2015.
- [57] Eli Goldberg, Norbert Driedger, and Richard I. Kittredge. Using natural-language processing to produce weather forecasts. *IEEE Expert*, 9(2):45–53, 1994.
- [58] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. *Numerische mathematik*, 14(5):403–420, 1970.
- [59] Arthur C Graesser, Shulan Lu, George Tanner Jackson, Heather Hite Mitchell, Mathew Ventura, Andrew Olney, and Max M Louwerse. Autotutor: A tutor with dialogue in natural language. *Behavior Research Methods*, 36(2):180–192, 2004.
- [60] Alex Graves. Sequence transduction with recurrent neural networks. *CoRR*, abs/1211.3711, 2012.

- [61] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.
- [62] Çağlar Gülçehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.
- [63] Çağlar Gulcehre, Marcin Moczulski, Misha Denil, and Yoshua Bengio. Noisy activation functions. In *International Conference on Machine Learning*, pages 3059–3068, 2016.
- [64] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 1321–1330, 2017.
- [65] Bikash Gyawali and Claire Gardent. Surface realisation from knowledge-bases. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 424–434, 2014.
- [66] Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.
- [67] Lushan Han, Abhay L Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. Umbc.ebiquity-core: semantic textual similarity systems. In *Second Joint Conference on Lexical and Computational Semantics (\*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, volume 1, pages 44–52, 2013.
- [68] Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on*

*Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4803–4809, 2018.

- [69] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.
- [70] Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. Scalable modified kneser-ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers*, pages 690–696, 2013.
- [71] Brent Hecht and Darren Gergle. The Tower of Babel Meets Web 2.0: User-generated Content and Its Applications in a Multilingual Context. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 291–300. ACM, 2010.
- [72] Michael Heilman and Noah A. Smith. Good question! statistical ranking for question generation. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA*, pages 609–617, 2010.
- [73] Sebastian Hellmann, Jens Lehmann, Sören Auer, and Martin Brümmer. Integrating NLP using linked data. In *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part II*, pages 98–113, 2013.
- [74] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010, Uppsala University, Uppsala, Sweden, July 15-16, 2010*, pages 33–38, 2010.
- [75] Daniel Hewlett, Alexandre Lacoste, Llion Jones, Illia Polosukhin, Andrew Fandrianto, Jay Han, Matthew Kelcey, and David Berthelot. Wikireading: A novel large-scale language understanding task over wikipedia. In *Proceedings of the*



*54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.

- [76] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR.
- [77] Heng Ji and Ralph Grishman. Knowledge base population: Successful approaches and challenges. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 1148–1158, 2011.
- [78] Thorsten Joachims. A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML 1997), Nashville, Tennessee, USA, July 8-12, 1997*, pages 143–151, 1997.
- [79] Ian T. Jolliffe. Principal component analysis. In *International Encyclopedia of Statistical Science*, pages 1094–1096. 2011.
- [80] Dan Jurafsky and James H. Martin. *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition, 2nd Edition*. Prentice Hall series in artificial intelligence. Prentice Hall, Pearson Education International, 2009.
- [81] Lucie-Aimée Kaffee. *Generating Article Placeholders from Wikidata for Wikipedia: Increasing Access to Free and Open Knowledge*. Bachelor’s thesis, HTW Berlin, 2016.
- [82] Lucie-Aimée Kaffee, Hady ElSahar, Pavlos Vougiouklis, Christophe Gravier, Frédérique Laforest, Jonathon S. Hare, and Elena Simperl. Learning to generate wikipedia summaries for underserved languages from wikidata. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New*

- Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 640–645, 2018.
- [83] Lucie-Aimée Kaffee, Hady ElSahar, Pavlos Vougiouklis, Christophe Gravier, Frédérique Laforest, Jonathon S. Hare, and Elena Simperl. Mind the (language) gap: Generation of multilingual wikipedia summaries from wikidata for article-placeholders. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, pages 319–334, 2018.
- [84] Lucie-Aimée Kaffee, Alessandro Piscopo, Pavlos Vougiouklis, Elena Simperl, Leslie Carr, and Lydia Pintscher. A Glimpse into Babel: An Analysis of Multilinguality in Wikidata. In *Proceedings of the 13th International Symposium on Open Collaboration*, page 14. ACM, 2017.
- [85] Mitesh M. Khapra, Dinesh Raghu, Sachindra Joshi, and Sathish Reddy. Generating natural language question-answer pairs from a knowledge graph using a RNN based question generation model. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 376–385, 2017.
- [86] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *(e)Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 - September 3 2004*, pages 180–191, 2004.
- [87] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
- [88] Ryan Kiros, Ruslan Salakhutdinov, and Richard S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *CoRR*, abs/1411.2539, 2014.
- [89] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics, 2003.

- [90] Ravi Kondadadi, Blake Howald, and Frank Schilder. A statistical NLG framework for aggregated planning and realization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 1406–1415, 2013.
- [91] Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc’Aurelio Ranzato. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5039–5049, 2018.
- [92] Rémi Lebret, David Grangier, and Michael Auli. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1203–1213, 2016.
- [93] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.
- [94] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia—a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 6(2):167–195, 2015.
- [95] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*, pages 333–342, 2017.
- [96] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*, pages 333–342, 2017.

- [97] William D Lewis and Phong Yang. Building MT for a Severely Under-Resourced Language: White Hmong. *Association for Machine Translation in the Americas, October, 2012*.
- [98] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain, 2004.
- [99] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 2181–2187, 2015.
- [100] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 2181–2187, 2015.
- [101] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.
- [102] Thang Luong, Ilya Sutskever, Quoc V. Le, Oriol Vinyals, and Wojciech Zaremba. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 11–19, 2015.
- [103] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.
- [104] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for*

- Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*, pages 55–60, 2014.
- [105] Diego Marcheggiani and Ivan Titov. Discrete-state variational autoencoders for joint discovery and factorization of relations. *Transactions of the Association for Computational Linguistics*, 4, 2016.
- [106] Teresa Martin, Fiete Botschen, Ajay Nagesh, and Andrew McCallum. Call for discussion: Building a new standard dataset for relation extraction tasks. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction, AKBC@NAACL-HLT 2016, San Diego, CA, USA, June 17, 2016*, pages 92–96, 2016.
- [107] Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. Open Language Learning for Information Extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP 2012*, pages 523—534. Association for Computational Linguistics, 2012.
- [108] David McClosky, Eugene Charniak, and Mark Johnson. Effective self-training for parsing. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 4-9, 2006, New York, New York, USA, 2006*.
- [109] Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. DBpedia Spotlight: Shedding Light on the Web of Documents. In *Proceedings of the 7th international conference on semantic systems*, pages 1–8. ACM, 2011.
- [110] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119, 2013.
- [111] George A Miller and Walter G Charles. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28, 1991.

- [112] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *the 47th Annual Meeting of the Association for Computational Linguistics (ACL' 09)*., pages 1003–1011. Association for Computational Linguistics, 2009.
- [113] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, pages 1003–1011, 2009.
- [114] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, pages 1003–1011, 2009.
- [115] Tom M. Mitchell, William W. Cohen, Estevam R. Hruschka Jr., Partha Pratim Talukdar, Justin Betteridge, Andrew Carlson, Bhavana Dalvi Mishra, Matthew Gardner, Bryan Kisiel, Jayant Krishnamurthy, Ni Lao, Kathryn Mazaitis, Thahir Mohamed, Ndapandula Nakashole, Emmanouil Antonios Platanios, Alan Ritter, Mehdi Samadi, Burr Settles, Richard C. Wang, Derry Tanti Wijaya, Abhinav Gupta, Xinlei Chen, Abulhair Saparov, Malcolm Greaves, and Joel Welling. Never-ending learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 2302–2310, 2015.
- [116] Thahir Mohamed, Estevam R. Hruschka Jr., and Tom M. Mitchell. Discovering relations between noun categories. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1447–1455, 2011.
- [117] Tal Montal and Zvi Reich. I, robot. you, journalist. who is the author? authorship, bylines and full disclosure in automated journalism. *Digital Journalism*, 5(7):829–849, 2017.

- [118] Yassine Mrabet, Pavlos Vougiouklis, Halil Kilicoglu, Claire Gardent, Dina Demner-Fushman, Jonathon Hare, and Elena Simperl. Aligning texts and knowledge bases with semantic sentence simplification, 2016.
- [119] Ndapandula Nakashole, Gerhard Weikum, and Fabian M. Suchanek. Discovering and exploring relations on the web. *PVLDB*, 5(12):1982–1985, 2012.
- [120] Ndapandula Nakashole, Gerhard Weikum, and Fabian M. Suchanek. PATTY: A taxonomy of relational patterns with semantic types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 1135–1145, 2012.
- [121] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: Building a Very Large Multilingual Semantic Network. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 216–225. Association for Computational Linguistics, 2010.
- [122] Axel-Cyrille Ngonga Ngomo, Lorenz Bühmann, Christina Unger, Jens Lehmann, and Daniel Gerber. Sorry, i don’t speak SPARQL: translating SPARQL queries into natural language. In *22nd International World Wide Web Conference, WWW ’13, Rio de Janeiro, Brazil, May 13-17, 2013*, pages 977–988, 2013.
- [123] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs: From multi-relational link prediction to automated knowledge graph construction. *arXiv preprint arXiv:1503.00759*, 2015.
- [124] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. Factorizing YAGO: scalable machine learning for linked data. In *Proceedings of the 21st international conference on World Wide Web*, pages 271–280. ACM, 2012.
- [125] Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. A survey on open information extraction. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 3866–3878, 2018.

- [126] Joakim Nivre. Incrementality in deterministic dependency parsing. In *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, pages 50–57. Association for Computational Linguistics, 2004.
- [127] Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan T. McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016.*, 2016.
- [128] Jekaterina Novikova, Ondrej Dusek, Amanda Cercas Curry, and Verena Rieser. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2241–2252, 2017.
- [129] Harinder Pal and Mausam. Donyms and compound relational nouns in nominal open IE. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction, AKBC@NAACL-HLT 2016, San Diego, CA, USA, June 17, 2016*, pages 35–39, 2016.
- [130] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.*, pages 311–318, 2002.
- [131] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global Vectors for Word Representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.
- [132] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543, 2014.



- [133] Yuval Pinter and Jacob Eisenstein. Predicting semantic relations using global graph properties. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1741–1751, 2018.
- [134] Yashaswi Pochampally, Kamalakar Karlapalem, and Navya Yarrabelly. Semi-Supervised Automatic Generation of Wikipedia Articles for Named Entities. In *Wiki@ ICWSM*, 2016.
- [135] Martin Potthast, Tim Gollub, Matthias Hagen, Johannes Kiesel, Maximilian Michel, Arnd Oberländer, Martin Tippmann, Alberto Barrón-Cedeño, Parth Gupta, Paolo Rosso, and Benno Stein. Overview of the 4th international competition on plagiarism detection. In *CLEF 2012 Evaluation Labs and Workshop, Online Working Notes, Rome, Italy, September 17-20, 2012*, 2012.
- [136] Rajat Raina, Andrew Y. Ng, and Christopher D. Manning. Robust textual inference via learning and abductive reasoning. In *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, pages 1099–1105, 2005.
- [137] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392, 2016.
- [138] Marek Rei and Ronan Cummins. Sentence Similarity Measures for Fine-Grained Estimation of Topical Relevance in Learner Essays. In *Proc. of the BEA Workshop 2016*.
- [139] Sebastian Riedel, Limin Yao, Benjamin M. Marlin, and Andrew McCallum. Relation extraction with matrix factorization and universal schemas. In *Joint Human Language Technology Conference/Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL '13)*, June 2013.

- [140] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III*, pages 148–163, 2010.
- [141] Bryan Rink and Sanda M. Harabagiu. A generative model for unsupervised discovery of relations and argument classes from clinical texts. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 519–528, 2011.
- [142] Michael Roth and Mirella Lapata. Neural semantic role labeling with dependency path embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.
- [143] Sebastian Ruder and Barbara Plank. Strong baselines for neural semi-supervised learning under domain shift. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1044–1054, 2018.
- [144] Vasile Rus and Mihai C. Lintean. A comparison of greedy and optimal assessment of natural language student input using word-to-word similarity metrics. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP, BEA@NAACL-HLT 2012, June 7, 2012, Montréal, Canada*, pages 157–162, 2012.
- [145] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 379–389, 2015.
- [146] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 379–389, 2015.

- [147] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 2988–2997, 2017.
- [148] Evan Sandhaus. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752, 2008.
- [149] Christina Sauper and Regina Barzilay. Automatically Generating Wikipedia Articles: A Structure-aware Approach. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1, ACL '09*, pages 208–216, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.
- [150] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083, 2017.
- [151] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.
- [152] Iulian Vlad Serban, Alberto García-Durán, Çağlar Gülçehre, Sungjin Ahn, Sarath Chandar, Aaron C. Courville, and Yoshua Bengio. Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.
- [153] Dominic Seyler, Mohamed Yahya, and Klaus Berberich. Generating quiz questions from knowledge graphs. In *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume*, pages 113–114, 2015.

- [154] Lifeng Shang, Zhengdong Lu, and Hang Li. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1577–1586, 2015.
- [155] Yatian Shen and Xuanjing Huang. Attention-based convolutional neural network for semantic relation extraction. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 2526–2536, 2016.
- [156] Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Y. Ng. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 935–943, 2013.
- [157] Anders Søgaard. Simple semi-supervised training of part-of-speech taggers. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden, Short Papers*, pages 205–208, 2010.
- [158] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.
- [159] Gabriel Stanovsky and Ido Dagan. Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2300–2305, 2016.
- [160] Gabriel Stanovsky, Ido Dagan, and Mausam. Open IE as an Intermediate Structure for Semantic Tasks. In *Association for Computational Linguistics*, pages 303–308, Beijing, 2015.
- [161] Gabriel Stanovsky, Jessica Fidler, Ido Dagan, and Yoav Goldberg. Getting more out of syntax with props. *CoRR*, abs/1603.01648, 2016.

- [162] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007.
- [163] Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 455–465, 2012.
- [164] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112, 2014.
- [165] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014.
- [166] Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. Probabilistic matrix factorization leveraging contexts for unsupervised relation extraction. In *Advances in Knowledge Discovery and Data Mining - 15th Pacific-Asia Conference, PAKDD 2011, Shenzhen, China, May 24-27, 2011, Proceedings, Part I*, pages 87–99, 2011.
- [167] Théo Trouillon, Christopher R. Dance, Éric Gaussier, Johannes Welbl, Sebastian Riedel, and Guillaume Bouchard. Knowledge graph completion via complex tensor factorization. *Journal of Machine Learning Research*, 18:130:1–130:38, 2017.
- [168] Joseph P. Turian, Lev-Arie Ratinov, and Yoshua Bengio. Word representations: A simple and general method for semi-supervised learning. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 384–394, 2010.

- [169] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2962–2971, 2017.
- [170] Shikhar Vashishth, Prince Jain, and Partha Talukdar. Cesi: Canonicalizing open knowledge bases using embeddings and side information. In *Proceedings of the 2018 World Wide Web Conference, WWW '18*, pages 1317–1327, Republic and Canton of Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee.
- [171] Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. RESIDE: improving distantly-supervised neural relation extraction using side information. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1257–1266, 2018.
- [172] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4566–4575, 2015.
- [173] Oriol Vinyals and Quoc V. Le. A neural conversational model. *CoRR*, abs/1506.05869, 2015.
- [174] Pavlos Vougiouklis, Hady ElSahar, Lucie-Aimée Kaffee, Christophe Gravier, Frédérique Laforest, Jonathon Hare, and Elena Simperl. Neural wikipedia: Generating textual summaries from knowledge base triples. *J. Web Sem.*, 52-53:1–15, 2018.
- [175] Pavlos Vougiouklis, Hady ElSahar, Lucie-Aimée Kaffee, Christophe Gravier, Frédérique Laforest, Jonathon S. Hare, and Elena Simperl. Neural wikipedia: Generating textual summaries from knowledge base triples. *CoRR*, abs/1711.00155, 2017.
- [176] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

- [177] Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. Relation classification via multi-level attention cnns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.
- [178] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pages 1112–1119, 2014.
- [179] Leo Wanner, Bernd Bohnet, Nadjat Bouayad-Agha, François Lareau, and Daniel Nicklaß. Marquis: Generation of user-tailored multilingual air quality bulletins. *Applied Artificial Intelligence*, 24(10):914–952, 2010.
- [180] Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.
- [181] Ji Wen. Structure regularized bidirectional recurrent convolutional neural network for relation classification. *CoRR*, abs/1711.02509, 2017.
- [182] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.
- [183] Michael J Wise. Yap3: Improved detection of similarities in computer program and other texts. *ACM SIGCSE Bulletin*, 28(1):130–134, 1996.
- [184] Minguang Xiao and Cong Liu. Semantic relation classification via hierarchical recurrent neural network with attention. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 1254–1263, 2016.
- [185] Shengwu Xiong, Weitao Huang, and Pengfei Duan. Knowledge graph embedding via relation paths and dynamic mapping matrix. In *Advances in Conceptual Modeling - ER 2018 Workshops Emp-ER, MoBiD, MREBA, QMMQ, SCME, Xi’an, China, October 22-25, 2018, Proceedings*, pages 106–118, 2018.
- [186] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell:

- Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 2048–2057, 2015.
- [187] Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. Semantic relation classification via convolutional neural networks with simple negative sampling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 536–540, 2015.
- [188] Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. Question answering on freebase via relation extraction and textual evidence. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.
- [189] Yan Xu, Ran Jia, Lili Mou, Ge Li, Yunchuan Chen, Yangyang Lu, and Zhi Jin. Improved relation classification by deep recurrent neural networks with data augmentation. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 1461–1470, 2016.
- [190] Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1785–1794, 2015.
- [191] Ying Xu, Mi-Young Kim, Kevin Quinn, Randy Goebel, and Denilson Barbosa. Open Information Extraction with Tree Kernels. In *The 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, HLT-NAACL 2013*, pages 868–877, Atlanta, Georgia, 2013.



- [192] Ying Xu, Christoph Ringlstetter, Mi-Young Kim, Randy Goebel, Grzegorz Kon-drak, and Yusuke Miyao. A Lexicalized Tree Kernel for Open Information Ex-traction. In *the 53rd annual meeting of the association for computational linguis-tics, ACL 2015*, pages 279–284, 2015.
- [193] Hiroyasu Yamada and Yuji Matsumoto. Statistical dependency analysis with sup-port vector machines. In *Proceedings of IWPT*, volume 3, pages 195–206. Nancy, France, 2003.
- [194] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embed-ding entities and relations for learning and inference in knowledge bases. *CoRR*, abs/1412.6575, 2014.
- [195] Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. Structured relation discovery using generative models. In *Proceedings of the 2011 Con-ference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1456–1466, 2011.
- [196] Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. Structured relation discovery using generative models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1456–1466. Associ-ation for Computational Linguistics, 2011.
- [197] Limin Yao, Sebastian Riedel, and Andrew McCallum. Unsupervised relation discovery with sense disambiguation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 712–720. Association for Computational Linguistics, 2012.
- [198] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network. In *COLING 2014, 25th In-ternational Conference on Computational Linguistics, Proceedings of the Confer-ence: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 2335–2344, 2014.

- [199] Lanjun Zhou, Binyang Li, Wei Gao, Zhongyu Wei, and Kam-Fai Wong. Un-supervised discovery of discourse relations for eliminating intra-sentence polarity ambiguities. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 162–171, 2011.
- [200] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*, 2016.
- [201] Zhi-Hua Zhou and Ming Li. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Trans. Knowl. Data Eng.*, 17(11):1529–1541, 2005.