GRAPH-BASED KNOWLEDGE REPRESENTATION FOR NATURAL LANGUAGE UNDERSTANDING

Submitted by
Gennaro Nolano

A thesis submitted for the degree of Doctor of Philosophy in
Literary, Linguistics and Comparative Studies

University of Napoli "L'Orientale"

Supervisor
Prof. PhD Johanna Monti
Coordinator of PhD Programme
Prof. PhD Rossella Ciocca

2021-2022

Abstract

In recent years the use of neural-based models trained on just textual data for specific tasks has become the go-to option in Natural Language Processing, both in academia and the industry. While these models show good results in specific frameworks, they present several issues, particularly in relation to data and model maintenance. These issues are made all the more difficult to solve given the inherent *black-box* aspect of neural models, which makes them particularly difficult to improve and update.
With regards to natural language understanding, an alternative to the textual-distributional representation implemented by neural models is represented by the use of structural, symbolic forms of representation for natural language utterances. These models are easier to implement and generally less data-hungry than neural frameworks, and can be implemented on top of neural models in order to solve some of their issues.
This work shows the implementation of such symbolic representation, in particular by implementing graph-based knowledge representation in three main tasks: semantic enrichment, relation extraction and question answering.
In particular, such representations are shown to be useful in connecting different sources of knowledge, namely texts and knowledge graphs available on the Web. By doing so, inherent contextual information extracted from texts can be integrated with common-sense and domain-specific knowledge in an explicit way. This is also beneficial since it creates an interpretable area that can act as a buffer against the black-box nature of neural models.

Preface

This thesis is the result of a three-year PhD Project developed in the Unior NLP Research Group at the University of Napoli "L'Orientale", Department of Literary, Linguistic and Comparative Studies.
This project has first and foremost been accompanied by two crucial periods abroad. The first one spanned from 01/12/2019 to 11/03/2020 and was spent at the *Adapt Centre*, in Dublin City University as part of the Erasmus+ Traineeship, under the supervision of two researchers, Eva Vanmassenhove and Guodong Xie. During this period, several techniques and research methods have been investigated, in particular in relation to the training and evaluation of neural models.
A second period abroad was spent at the University of Bielefeld, under the supervision of professor Philipp Cimiano (Semantic Computing Group), where the research project and the structure of the present work has been refined. Furthermore, this period abroad allowed for the definition of future projects regarding the use of graph-based structures for natural language understanding.

Projects & Collaborations The PhD period saw the involvement in several projects. While most of them are described in the present thesis as related to the general topic of natural language representation, several other collaborations have been carried out in other fields as well.
The first one of these collaborations is Spotto-IT[1], a project focusing on the investigation of university students' speech, in particular during the lockdown period in Italy due to COVID-19 policies. In the course of this project, I was involved with the recollection, maintenance and visualization of the data.

Similarly, I was involved in the generation of the ITAGlam corpus[2], a collection of tweets published by Italian Galleries, Libraries, Archives and Museums

[1] https://sites.google.com/view/unior-nlp-research-group/blog#h.hwo5hl36r7dg
[2] https://www.sites.google.com/view/unior-nlp-research-group/glob#h.ecyhi53zqlz

(GLAMs), during the lockdown period in Italy. This corpus is annotated according to communication strategies.

Finally, I was involved in the "VICINA" project[3], a virtual assistant which offers the opportunity to take a virtual tour guided by a conversational agent, and to explore the map of China created in 1719 by Matteo Ripa and kept in the Museo Orientale Umberto Scerrato of the University of Naples "L'Orientale".

Awards

- 3rd place at the EU DATATHON 2022[4] with the proposal Across Europe With MAGGIE, described in Section 5.3;
- Selected presentation during the Conference on the Future of Europe (COFE) Datathon[5] with a project dealing with multilingual keyphrase extraction and topic clustering, described in Section 3.3;
- First place at the Challenge for Europeana AI/ML Datasets[6] with the development of the Named Entity Recognition in Archaeological Texts (NEAT) dataset, as described in Section 3.2.

Publications

- Nolano, G., di Buono, M. P., Monti, J. (2022). From Monolingual Multiword Expression Discovery to Multilingual Concept Enrichment: an Ontology-based approach. Computational and Corpus-based Phraseology, 28, 197.
- di Buono, M. P., Gonçalo Oliveira, H., Barbu Mititelu, V., Spahiu, B., Nolano, G. (2022). Paving the way for enriched metadata of linguistic linked data. Semantic Web, (Preprint), 1-25.
- Carlino, C., Nolano, G., di Buono, M. P., Monti, J. (2021). # LaCulturaNonsiFerma: Report on Use and Diffusion of # Hashtags from the Italian Cultural Institutions during the COVID-19 outbreak. arXiv preprint arXiv:2103.11865.

[3]https://sites.google.com/view/unior-nlp-research-group/blog#h.g7fz0fg3ivlt

[4]https://op.europa.eu/en/web/eudatathon

[5]https://futureu.europa.eu/en/pages/datathon?format=html&locale=en

[6]https://pro.europeana.eu/post/europeanatech-challenge-for-europeana-ai-ml-datasets-announcing-the-winners

- Chiusaroli, F., Monti, J., Pierucci, M.L., di Buono, M.P., Nolano, G. (2021). Il corpus Spotted-Poivorrei-Ita:: la comunicazione del COVID-19 nella scrittura degli studenti universitari. Elementi per la sentiment analysis. Rassegna Italiana di Linguistica Applicata, 53(3), 209-225.
- Nolano, G., Elahi, M. F., di Buono, M. P., Ell, B., Cimiano, P. (2021). An Italian Question Answering System Based on Grammars Automatically Generated from Ontology Lexica. In CLiC-it.
- Monti, J., Chiusaroli, F., Pierucci, M. L., Nolano, G. (2021). # poivorrei tornare in aula: viaggio sentimentale negli atenei italiani in tempo di covid-19: un corpus linguistico e un caso studio di etnografia della comunicazione tra studenti universitari. # poivorrei tornare in aula: viaggio sentimentale negli atenei italiani in tempo di covid-19: un corpus linguistico e un caso studio di etnografia della comunicazione tra studenti universitari, 137-163.
- Carlino, C., Nolano, G., di Buono, M. P., Monti, J. (2020). LaCulturaNonSiFerma–Report su uso e la diffusione degli hashtag delle istituzioni culturali italiane durante il periodo di lockdown. arXiv preprint arXiv:2005.10527.
- Chiusaroli, F., Monti, J., Pierucci, M. L., Nolano, G. (2020). "Spotto la quarantena": per una analisi dell'italiano scritto degli studenti universitari via social network in tempo di COVID-19. Computational Linguistics CLiC-it 2020, 106.
- Nolano, G., Carlino, C., di Buono, M. P., Monti, J. (2020). ItaGLAM: A corpus of Cultural Communication on Twitter during the Pandemic. Computational Linguistics CLiC-it 2020, 323.
- Nolano, G., di Buono, M. P., Monti, J. UNIOR NLP@ COFE Datathon 2022 Task 2-Idea clustering.
- Nolano, G., di Buono, M. P., Monti, J. UNIOR NLP@ COFE Datathon 2022 Task 1-Keyword extraction from proposals.

Contents

Part I

Natural Language Understanding

Chapter 1

Introduction

1.1 What is Natural Language Understanding

Natural Language Understanding (**NLU**) is, together with Natural Language Generation (**NLG**), one the main subareas of Natural Language Processing (**NLP**). While NLG is generally concerned with the automatic production of new utterances by machines, NLU has at its core the automatic *comprehension* (however one may define such concept) of utterances created by human speakers (Allen, 1987). This task is, on one hand, extremely difficult to tackle in terms of its practical objectives (first and foremost the definition of what would it mean for a machine to understand human language is still debated), but, on the other hand, its practical applications are many, and its study has important undertones in defining how humans understand language (Bender et al., 2020).

This introductory chapter is focused on the general definition of NLU as a field of research, and on the definition of a framework for natural language representation.

The first section is dedicated to the different definitions of language understanding that have been given through the years, mainly with regards to two points of view: philosophy of language and machine understanding. Then, it describes the most recent practical applications of NLU.

The second section describes the desiderata for the representation of natural language in a machine-readable format, together with the linguistic challenges inher-

ently present in such a process.
The third section covers the general topic of text-based representations, with a focus on Large Language Models (LLMs) and on the issues related to this kind of representation models, while the fourth section introduces graphs as a means to represent knowledge and language, and frameworks for graph representation.
Finally, sections 5, 6 and 7 describe the dissertation in terms of research questions, contributions to the field and outline.

1.1.1 Criteria of Understanding

While the question of what would it mean for a machine to be able to *understand* a natural language utterance has a long history on its own (M. R. Quillian et al., 1962), before recent efforts philosophers of language, linguists, logicians and philosophers have all proposed several semantic theories aiming at describing what it means, for human speakers, to understand a piece of human-generated communication (Allen, 1987).
In recent years, the goal has rather been moved to what it would take for a machine to act as if it understood a piece of human-generated communication in a human-like fashion.
The following paragraphs are dedicated, respectively, to classical approaches to language understanding[1], and to the more recent approaches to this topic with regards to machine understanding.

Linguistics and Philosophy of Language One the most important themes of natural language understanding from a human point of view is, historically, its connection to the major philosophical concept of *truth* (Kölbel, 2008; Tarski, 1944b). More specifically, this is related to the question of what does it take for a sentence to be considered *true*, and what are its *truth conditions* (i.e. the conditions according to which a sentence can be defined as being true). This is the core idea behind classical theories of semantics.
These questions have a long history, and one of the first attempts of defining meaning from this point of view is found in the works of ancient Greek philosophers, in particular Aristotle.
According to Aristotle, the truth of a sentence does not lie in the sentence itself,

[1]An analysis of semantic theories' evolution and their interconnection is beyond the scope of this paper, and the interested reader is referred to Maienborn et al. (2019) and Speaks (2021).

neither does it lie in the objects it describes, but rather in a general property of the sentence defining its relation to the objects described (Wheeler, 1999). Based on this assumption, the first step in understanding a sentence is its grounding to the concepts and objects it refers to (its referents).

In this sense, a sentence such as *"Leonardo da Vinci painted the Mona Lisa"* is true since its references in reality hold the same relations they hold in the sentence. Since the same cannot be said of *"Leonardo da Vinci painted the Nightwalkers"*, this sentence is false, and this is the case because humans understand both the referents described in these sentences, and the properties expressed in them.
A formalisation of such a theory of reference, and in particular on its distinction from the concept of sense (i.e., the meaning expressed by a word, even without a real-world reference), can be found in the works of Frege (1948).
Fregean distinction between sense and reference has had a crucial influence in modern semantics, particularly as a framework for the definition of the issues related to the general theory of reference. In particular, it is useful in defining those situations in which references themselves might not be known to the speaker.
This is the case, for instance, of sentences such as *"the tallest man in the world"*, which is understandable even without knowing what pieces of reality it directly refers to.
Carnap (1947) focus on the proposal of a methodology to solve these situations, with the core idea being that to understand an utterance does not mean to know the referent itself (as it might not even exists in our world), but rather to understand it as a *function*, which can point to a (possibly unknown, or even empty) reference.
These functions are what Carnap (1947) calls intensions, which are evaluated according to context-specific circumstances and world knowledge. This idea of a function representing the sense of an utterance, which is understood as separated from the conditions which would make it true, is the core idea behind *possible world semantics* (Lewis, 1970).

One of the first modern attempts of a structured framework for language understanding behind the classical scope of reference was the *truth-conditional semantics*, proposed by Davidson (1967). This theory has as its foundation Tarksi's semantic theory of truth (Tarski, 1944a), which attempts to formally define what criteria a utterance should meet in order to be considered true. One of the most influential proposals by Tarski is the implementation of a *metalanguage* to define the truth conditions of a utterance in a given language in order to avoid paradoxes, which anticipates the implementation of logical languages to represent utterances' meaning.
Tarski is mainly interested in defining the afore-mentioned truth criteria, and states

that a set of meta-sentences defining the truth conditions of every possible sentence in a given language would give us a full interpretation of said language.
Davidsonian semantics, on the other hand, further states that such recollection of metasentences would, in and by itself, determine the full *understanding* of the language, without any need for explicit connection between sentences and references, which is why this approach is referred to as *truth-conditional semantics*. For further discussion on the topic, and critiques on the framework, see Burge (1986), Davidson (1976), Larson et al. (1993), Lepore et al. (1989), Schiffer (1987), and Soames (2002).
Several aspects of truth-conditional semantics were later employed in Montague grammar (Montague, 1970a). In particular, one of the main points defined in Montague's work is the importance of compositionality of the syntax-semantics interface, that is how the meaning of a larger linguistic unit can be built up on the basis of its elements and their syntactic structure. Such compositionality plays an extremely important role in NLU in general, and in particular in the context of semantic parsing (Bender et al., 2015).
Montague further amplifies the scope of this paradigm by integrating it with the notion of *contextual parameters* (Montague, 1970b), underlying the importance of the context of use of utterances in understanding their meanings.

An alternative to truth-condition semantics is what is generally referred to as *inferentialism*, which formalises meaning understanding through the use of inferences and of entailments of a given statement, rather than its references and truth-conditions (R. Brandom, 2010). While this paradigm does not fully deny the assumption that semantic theory should be invested in the definition of truth conditions, it denies their fundamental role.
Inferential semantics has been defined by R. B. Brandom (2000), who underlines that the notion of truth follows the notion of good and bad inferences. This means that, rather than being interested in what a piece of human communication refers to in the real world, an inferentialist would be more interested in the inferences that can be drawn starting from the same piece of communication.
This directly leads to the idea that the meaning of sentences cannot be retrieved by looking at independent utterances and their connection to the world, but that meaning lies, instead, in the inherent relations between sentences themselves.
One of the main features of the previously defined semantic theories is that the main focus of the analysis lies in utterances, and not much has been said about actual discourses. As a way of covering conversations, a semantic theory by the name of dynamic semantics was developed (Heim, 2012; Kamp, 1981), with the objective of approaching semantics in the sense of contextual changes, that is how

discourse is updated in the course of a conversation. This can be seen as a sort of "extension" of previous theories, rather than being in contrast with any of them (Stojnić, 2019).
One of the main applications of this theory is Discourse Representation Theory (DRT), as defined by Kamp (1981). The key component of DRT is a discourse representation structure (DRS), which is a mental representation of a discourse, dynamically built up as the hearer receives more information.

Machine Understanding Over the years the focus of natural language understanding shifted more towards machine understanding of natural language, which is interested in what it would mean for a machine to understand natural language. This question is characterized by several approaches, each defining specific issues and specific possible goals.

One of the earliest and most influential efforts in this regard is the Turing test (Turing, 1950), which also ended up being one of the most popular ways of describing how a theoretical *human machine* would act. According to this test, a machine could be said to be able to *think as a human* (and, in turn, understand human language) if a situation arises in which a human could not be able to distinguish such a machine from a human interlocutor.
The main idea behind the Turing test is that the question to be asked is not what it would mean for a machine to think, but rather how such a machine *would act*. This is particularly beneficial since actual behaviors can be seen and evaluated, while inner mind processes are generally difficult to analyse and emulate. According to Turing, thus, the interactive use of language would be the most important task to aim at in order to build an understanding machine. This paradigm is also known as *behaviouralism*.

One main point of criticism towards this specific kind of behaviouralism applied to machines is that humans tend to be extremely quick at assigning intelligence to inanimate objects. This is true for computers as well, as it was proven in the ELIZA experiments, with what would later come to be known as the ELIZA effect (Block, 1981; Weizenbaum, 1966, 1976).
While ELIZA is a simple program based on pattern recognition and question rephrasing, poeple involved in the experiments were extremely invested in their conversations with the machine, with many of them ending up talking about their personal issues and experiences as they would do with a human speaker.
This was particularly true for the DOCTOR script in ELIZA, which simulates a Rogerian pyschotherapist (Hutchens, 1996). Nevertheless, ELIZA shows no sign

of actual understanding, being based on a simple system of pattern recognition and rephrasing.

Over the years, this kind of behaviouralism has also been criticized as being incomplete, since the use of language on its own cannot prove by itself the presence of actual understanding. For instance, according to McCarthy (1968) the way knowledge is represented should also be taken into account in machine understanding. The author, in particular, focuses on how symbolic representations are derived, and on the interconnections between these derivations and specific actions from the machine.

One further development away from behaviouralism is the shift towards the *means* through which use and understanding of language has been obtained, as it was also famously described by Searle (1980) in the Chinese Room experiment.

According to this thought experiment, while one could be able to build a computer that can answer every question asked in Chinese, while also being able to pass the Turing test, the question arises on whether this computer does actually understand Chinese, or it just simulates this ability by mimicking the language through data learned. These two opposite positions are defined by the author as *strong AI* and *weak AI*, respectively.

Given this framework, Searle deems the strong AI hypothesis as false, since a human agent would be able to replace the computer and follow its instructions step-by-step, even while not being actually able to speak Chinese. Thus, the author concludes, one must assume that the use of language is not actual proof of understanding.

It is thus clear that the understanding of utterances by humans is only *part* of the whole issue. Language should also be linked to actual knowledge and experience of the world, lest be an incomplete form of machine understanding. In relation to language understanding, this is further made clear by Harnad (1990), who shows the impossibility of learning one's native language by using just a dictionary.

More recently, Levesque (2014) investigates the correct questions researchers should ask themselves when dealing with understanding for AI. That is, while behaviour is the main focus of research, one should also answer what this behavior depends on, and how the final behaviour is achieved.

In particular, the author focuses on how language and knowledge representation should go in parallel in order to have a real understanding.

Similar issues have had a practical impact in the field of Natural Language Understanding, in particular with a raise in awareness in relation to the true capabilities of recent NLU systems. This topic is the main focus of the discussion in Section 1.3.

Recent Implementations While in recent years the discourse on NLU has rather shifted towards machine understanding, it is interesting to notice how classical theories of semantics, while often not being explicitly recognized, have been implemented in various NLU-related tasks.
The objectives of truth-conditional semantics, for instance, are extremely similar to the task of Common sense reasoning, where a model is required to predict whether a utterance should be considered true or not[2].
Inferentialism, on the other hand, shows many similarities with the task of Natural Language Inference (NLI), which is interested in whether an hypothesis is true, false, or neutral given a premise[3]. It could also be argued that the task of Next Sentence Prediction, one of the two core tasks of the pre-training of the popular Language Model Bidirection Encoder Representation from Transformers (BERT) (Devlin et al., 2019), has its roots in this semantic theory.
Furthermore, the tasks of Entity Recognition[4], Relation Classification[5], Word Sense Disambiguation[6] and Semantic Parsing in general can be traced back to questions concerning reference and sense.
Finally, the importance of keeping track of the flow of discourse, highlighted by DRT, is of extreme importance in the general topic of conversational agents, and more specifically in the task of dialogue state tracking[7]. This task aims at keeping a state of the dialogue, according to the information dynamically inserted by an user in the course of a conversation. Similarly, the use of DRT for question representation in QA been explored in depth by Cimiano et al. (2014).

1.1.2 Applications

Despite the ongoing philosophical debates regarding what does it means to *understand* natural language (for both humans and machines), over the years many applications have been developed for the automation of NLU, with new states of the art being established extremely fast, and models becoming obsolete extremely quickly (Hershcovich et al., 2021).

[2]http://nlpprogress.com/english/common$_{s}$ense.html
[3]http://nlpprogress.com/english/natural$_{l}$anguage$_{i}$nference.html
[4]http://nlpprogress.com/english/named$_{e}$ntity$_{r}$ecognition.html
[5]http://nlpprogress.com/english/relationship$_{e}$xtraction.html
[6]http://nlpprogress.com/english/word$_{s}$ense$_{d}$isambiguation.html
[7]http://nlpprogress.com/english/dialogue

The main reasons for this *rapid hill-climbing* (Bender et al., 2020) is that the commercial applications of NLU models are many and extremely remunerative and that, as a result, long-term goals (such as the one proposed by semantic theories) have been abandoned in favor of short-term increase in performance.

This section describes some of these applications of particular interest for recent development in the field, namely applications for man-machine interaction. While direct interaction with a machine generally calls for the use of some sort of artificial language, NLU models avoid the need to learn and master such languages by making machines responsive to natural-language utterances.

Question Answering

One of the most important applications for NLU is the creation of systems that can automatically answer questions posed in natural language. Question Answering (QA) has been one of the earliest applications of NLP, and as early as the 1960s, A. V. Phillips (1960) describes one of the first approaches to QA, based on question parsing and text matching on unstructured documents.

Models for QA usually fall into one of two categories: Knowledge-based QA (KBQA) and Information Retrieval (IR) systems. While both models take a natural language question as input and give a natural language answer as output, the former retrieve the answer from a structured Knowledge Base, while the latter is interested in retrieving answers from a textual source, such as an article or a paragraph in a document.
This affects the way in which questions are internally represented by the system: questions posed to a KBQA are usually first mapped to a symbolic representation (e.g., queries that can be used with the background KB); while IR systems generally make use of distributional representations and similarity measures.

One of the earliest examples of KBQA is represented by the BASEBALL system (Green et al., 1961), in which questions get mapped to a formal representations by reformulation using domain-specific templates, and answers retrieved by filling the needed slots in the templates, as shown in Table 1 for the question *"Where did the Red Sox play on July 7th?"*.
The BASEBALL system showed the applications of *semantic parsing* (i.e., the explicit mapping of a natural language utterance to a formal representation), which has since then become a standard operation in KBQA.
Nevertheless, the model imposed many restrictions on the types of questions that could be asked: only single clauses could be used, logical connectives were pro-

`Place:`	?
`Team:`	Red Sox
`Month:`	July
`Day:`	7

Table 1: Example of a reformulation done by the BASEBALL system from Green et al., 1961. The system knows it has to answer for the value `Place`.

hibited, and the system made use of an exact matching to retrieve entities, which meant that only canonical forms could be used, without any space for entity disambiguation.

Over the years many other models have been built for KBQA, with early attempts focusing on answering simple questions involving a single *relation* and having a single *entity* as the answer (Bordes et al., 2015; Dong et al., 2015), while more recently the focus has shifted to the answering of *complex questions* containing multiple entities or expressing compound relations (S. Hu et al., 2018; Luo et al., 2018).
Furthermore, in recent years the rise in popularity of Knowledge Graphs for knowledge representation has led to development of models answering questions over Knowledge Graphs, in what is generally known as Knowledge Graph Question Answering (KGQA) (Y. Sun et al., 2019; Y. Zhang, Dai, et al., 2017).
Since entities in KGs are inherently interconnected, the use of such a knowledge base makes it possible to create explicit reasoning paths between distant entities (i.e., entities that are not directly connected, but which can be linked passing through multiple relations).
This process, known as *multi-hop reasoning*, is currently the focus of many research efforts (Lv et al., 2021). Furthermore, the use of graphs makes it so that specific graph-based algorithms (e.g. shortest paths and random walks) can be implemented, increasing the performance of the task.

IR models, also known as *text-based systems* in earlier surveys (Simmons, 1965), make use of information from an unstructured source of data (e.g., a document in natural language) to answer a question posed by an user. One of the earliest examples of this paradigms in the Protosynthex model described by Simmons et al. (1964), which exploits dependency parsing as a mean to retrieve useful spans of text regarding specific entities of interest.
Nowadays, such systems generally make use of contextual word embeddings trained on a reading comprehension dataset (e.g. SQuAD 2.0 Rajpurkar et al., 2018, or CliCR Šuster et al., 2018), in which questions are paired with a passage

from a paragraph containing the answer, and with the span of text representing answer itself.
These systems usually make use of an internal distributional representation for words, such as BERT embeddings, without explicit knowledge integration (C. Zeng et al., 2020). Since word embeddings can be used to train such a system autonomously, they do not need an explicit step of semantic parsing.

Conversational Agents

While QA systems focus on the specific task of answering a given question, Conversational Agents (also called Chatbots) are more general systems designed to emulate general conversations with human interlocutors.
These systems have become extremely popular in recent years, but their roots are found in an early system called ELIZA (Weizenbaum, 1966).
As previously mentioned, ELIZA is based on a pattern matching methodology which creates the illusion of understanding human language. The patterns refer to scripts, which specify rules and directions to follow on the basis of some given *persona* that the system can imitate.
For instance, the `DOCTOR` script focuses on giving advice on mental health by simulating a Rogerian psychotherapist, and one of its patterns is shown in Table 2.
While these results might easily fool an human interlocutor, and even pass the

User	**ELIZA**
I am X	How do you feel about being X?
I am unhappy	How do you feel about being unhappy?

Table 2: Example of a pattern from ELIZA.

Turing test (Hutchens, 1996), it is clear that they act only on the basis of linguistic patterns, and no actual knowledge of the world is involved in the process.

A more recent model for conversational agents is the corpus-based dialogue system defined by Inui et al. (2003), in which input sentences are first parsed using phrase-patterns and N-grams, and then information extracted in this way is used to find the most similar dialogue from a background dialogue corpus. Finally, information from this background dialogue corpus is used to generate responses by generalizing over possible templates.
While this model is a step towards the integration of linguistic information, it still fails to fully integrate knowledge about the world. Furthermore, since it relies on

templates from a background dialogue corpus, it does not generalize over all the possibilities of human language.

Nowadays, chatbots have become extremely complex, and capable of responding to several different situations. The online application Cleverbot[8], for instance, responds to natural language input by comparing the current conversation to all the ones that have happened in the past. By constantly updating its own background dataset, Cleverbot is able to adapt to unknown situations. Nevertheless, semantic information are still not explicitly stated in the model.

A more recent chatbot is represented by the OpenAI system ChatGPT[9], which makes use of a pre-trained large language model fine-tuned for conversational purposes. Its results are extremely impressive, but the responses have been conflicting (Gozalo-Brizuela et al., 2023; Pavlik, 2023; Susnjak, 2022).

Some Conversational Agents have also found more common uses in our daily lives: for instance, it is the case of voice-based virtual agents, such as Siri[10], Google Assistant[11], and Cortana[12]. In order to answer for specific needs (e.g., playing music, making appointments, etc.) these systems make use of external datasets and complex inner representations to complete specific tasks while keeping track of dialogues.

Robot Navigation

The task of robot navigation focuses on the development of robots that are able to physically move in a spatial environment on the basis of directions uttered in natural language (Gul et al., 2019). This task is made particularly difficult by how spatial relations are represented in natural language, and in particular by the use of specific landmarks and relative directions for space points.

Traditionally speaking, the efforts in this area of study have produced two main approaches to the task of representing space (Thrun, 1998): grid-based paradigm (Elfes, 1987; Moravec, 1988), which represents space as a grid containing possible objects or obstacles, and topological paradigm (Golfarelli et al., 2001; Kortenkamp et al., 1994), which represents space as a graph, where each node corresponds to distinct situations, places or landmarks, and nodes are connected in

[8] https://www.cleverbot.com/

[9] chat.openai.com

[10] https://www.apple.com/it/siri/

[11] https://assistant.google.com/

[12] https://www.microsoft.com/en-us/cortana

case a walkable path exists between them.
While these models are useful in defining space, they represent no knowledge about the objects and places encountered, which makes the integration of such systems together with natural language commands particularly difficult. An advance in comparison to these traditional implementations is to represent space in what is generally called a *semantic map* (Nüchter et al., 2008).
The main idea behind a semantic map is to represent space in the form of places (e.g., a kitchen, the bedroom), objects (e.g., chair, bed) and the relations between them (Crespo et al., 2020) which are independent of the map content, but can be used for reasoning by the machine (Nüchter et al., 2008).
This knowledge can then be used as context by the machine when reacting to natural language utterances, and it is particularly useful when the robot has to react to some new objects its sensors have never seen before (Kollar et al., 2009).
While the acquisition of a semantic map can be made automatic, it can also benefit from the assistance of human users, and the integration of natural language in this field has also been recently explored (Crespo et al., 2017).
A background semantic map can then be used as a Knowledge Base on which natural language instructions can be grounded. For instance, in Thomason et al. (2015), human instructions are first mapped to a logical form representation through semantic parsing (see Section 2.2), and then grounded to a conceptual representation of the environment, in the form of a manually-created semantic map.
In Shah et al. (2018), on the other hand, instructions are implicitly connected to the visual representation of the agent's observations. These observations make no use of an explicit source of background knowledge, but are fed into a neural architecture which attempts at recognizing specific objects in what the machine is visualizing.
In this way, while no semantic map is explicitly used, the agent can adapt to what exists in front of it. The main advantage to this method is that it can more easily adapt to new instructions and to environmental changes.

1.2 Defining Representation

In recent years efforts regarding human-machine interactions have shifted the attention to a more practical approach, in particular in relation to the oft-stated goal of NLP to build "human-like" models (Eger et al., 2019; Hershcovich et al., 2021;

Linzen, 2020; Tamari et al., 2020; J. Wang et al., 2019).
In this sense, it is important to note once again that, while the philosophical discussion on what it means to understand a sentence is long and still alive, many applications have been built on the basis of the semantic theories previously described.
The main way through which these semantic theories are integrated in actual applications is through the integration of *semantic representation* of utterances so that natural language can be made *understandable* (i.e., parsable and automatically usable) by machines, so that certain actions can be taken as a response.
The following sections take a look at the desiderata of such a representation, and then describe some of the linguistic challenges inherently present in the mapping of natural language utterances into machine-readable data formats.

1.2.1 Desiderata for Representation

The *desiderata* for the kind of semantic representation described in this work have been laid out by Allen (1987), who defines a representation as one solving two important issues in the automatic process of natural language understanding.
In particular, Allen refers to two separate, but not independent, problems:

1. Words and sentences ambiguity:
    - for each meaning of an ambiguous linguistic unit, at least one possible representation should be retrieved, and this representation should be different from that of any other meaning of the same ambiguous linguistic unit;
    - linguistic units with similar meanings should have similar representations.
2. The explicitation of the knowledge needed to fully understand an utterance and its implicatures.

In Allen's work these two issues are then related to two different representations, respectively logical forms and world representations. While logical forms are derived by the disambiguation of small linguistic units and their recombination to produce the interpretation of larger phrases, world representations refer to the context in which these linguistic units exist, and according to which they should be interpreted.
While many kinds of semantic representations have been proposed in recent years (see Section 2.1), they all refer in one way or another to the features hereby defined, with differences in how they realize their final output, and the source of

knowledge used to retrieve representations.

1.2.2 Linguistic Challenges

While the definition of a coherent method for semantic representation is of critical importance for many NLP tasks, many issues are inherently present in the systematization of natural languages, and they affect how machines should handle specific linguistic phenomena. Such phenomena are present at all linguistic levels, and they will be the focus of the following sections.

Syntax-semantics interface

As mentioned before, the interaction between syntax and semantics is a particularly relevant topic in the literature of semantic theories, ever since classic theories.

While certain syntax-semantics interaction patterns are easy to handle and predict (Shi et al., 2020), some syntactic phenomena are particularly hard to tackle for automatic models with regards to the meaning they express.

One of such phenomena is the handling of long-term dependencies. Dependencies in general are very important in terms of syntax-semantics interaction, since it can be generally assumed that a dependency relation between two words also implies the presence of a semantic relationship between the two (Temperley et al., 2018). Furthermore, it has been proven that the dependency structure of a sentence heavily affects its processing difficulty by human users (Gibson, 1998, 2000), in particular for what regards long-distance dependencies (C. Phillips et al., 2005), which is the case, for instance, of the object-verb relation in sentences such as *"The captain who the sailor greeted is tall"* (Piñango et al., 2016).

The same has been observed with machines, for which the issue of long-distance dependency relations has been the focus of many studies with regards to Neural Networks (Bengio et al., 1993; Bengio et al., 1994) and the general topic of vanishing gradients (Bhatt et al., 2020; Le et al., 2016).

The main source of this problem lies in the inherent lack of an upper bound on the number of sentences that a natural language can generate, as underlined by Chomsky (1956) in his investigation on the recursiveness of language. Thus while humans are capable (with a certain degree of difficulty) of understanding sentences made up of an ever-increasing number of words, the same cannot be said about machines, for which retaining information spanning over long time periods is a difficult task. For more on the topic, see Chomsky (1957) and Roeper (2010)

Another important issue in the field of syntax-semantics interface has to do with coreference and anaphora resolution.
While coreference resolutions deal with the grouping of an entity and all its mentions referring to it in a text, anaphora resolution deals with the solving of such references within the text with the same sense on a linguistic level. The difference between the two is well defined by the example proposed by Mitkov (2001): *Every speaker had to present his paper*. While we can say that "his" and "every speaker" are in an anaphoric relation, since in the context of this sentence they both refer to the same entity, they are not co-referent: we would never say that the sentence means that *every speaker had to present every speaker's paper*, since they refer to two different extra-linguistic entities. This is also closely related to the more general issue of scope ambiguity and quantification (Montague, 1974).
These two tasks have been the focus of many studies and efforts in the NLP literature (Sukthanker et al., 2020), with the main issues being, once again, that anaphoric elements may happen distant from each other, and the many different forms that the references can take, a list of which can be found in Sukthanker et al. (2020). Some of these references pose particular difficulties to machines, for instance zero anaphora (Iida et al., 2006) and split anaphoras (J. Yu et al., 2021).

Semantics

Some issues present in natural language representation are linked to phenomena inherently present in natural language semantics. One of the central problems in this field is ambiguity (MacDonald et al., 1994), as already mentioned in relation to Allen's desiderata for representation. As defined by Pinkal (1995), a sentence can be seen as *semantically indefinite* (i.e., ambiguous) when, despite sufficient knowledge of the relevant facts described by the sentence itself, it is not possible to assign to it a clear truth value.
The simplest level of ambiguity is found at the level of lexicality, which deals with the possible meanings that a linguistic unit can have, according to the context it is used in (MacDonald et al., 1994). Words, for instance, might have different meanings, and the same meaning can be realized by different surface forms. This issue is generally tackled by tasks such as word sense disambiguation[13], and it is of particular importance in the context of contextual word embeddings.

[13] http://nlpprogress.com/english/word$_s$ense$_d$isambiguation.html

While words can be ambiguous on their own, another level of ambiguity is found in the way linguistic units are combined according to the principle of compositionality (Heim et al., 1998). An example of such ambiguity arises from the previously mentioned scope ambiguity (Musolino, 2006): in the sentence *Every horse didn't jump over the fence* (Scontras et al., 2021), it is not clear whether *every* refers to *none* of the horses, or to *all except some* of the horses. More formally, in mapping this sentence to a logical form, it is not clear whether the quantifier *every* should be translated as $\forall$ or $\exists$.
The discourse on compositionality has been extensively addressed by authors such as Frege (1948), Montague (1970a), and Partee (2010).

Another issue in relation to semantics is the handling of unseen language units. For instance, recognizing and dealing with new words has been an important topic of discussion in recent works on NLU (Cui et al., 2021; Schick et al., 2020), with recent distributional models poorly handling rare words (Gong et al., 2018; Ruzzetti et al., 2022; Q. Wu et al., 2020; W. Yu et al., 2021).

Pragmatics

Any real-world process related to natural language is inherently interactive. Humans acquire language through interactive feedback (Thornton, 2008), and in a human-to-human environment language generation and understanding is constantly adjusted according to the surroundings (Labov, 2001). While this is extremely natural for humans, the same cannot be said about machines.
The effects of the inherent difficulty in dealing with pragmatics on a processing level are twofold: on one hand, it heavily affects human-machine interaction, in particular in how humans react to machines' violation of cooperative principles (Jacquet et al., 2021; Saygin et al., 2002); on the other hand it affects how machines deal with utterances in a real-world settings, which might have negative effects on the final application.
Generally speaking, the main intuitions about pragmatic theory are due to Grice (1989), who stated the central role of common sense reasoning about the intents of the speaker in communication.
From this point of view, participants of a conversation actively seek an explanation for the choices of the other participants, and, in doing so, they fill the blanks left out by the speakers, on the basis of shared knowledge. This is generally defined as cooperative principle.
This kind of shared knowledge is not inherently present in machines, and it has

to be explicitly delineated through the definition of specific knowledge bases and ontologies, on which inferences can be made.
In absence of such knowledge, it is extremely difficult to train for a system given that certain inferences might be never observed in the training set, which makes it difficult to compute their distribution. This might be the case, for instance, of causal inferences (Feder et al., 2022; Holland, 1986).
While such knowledge bases can be represented by actual structures, such as semantic maps (previously mentioned in Section 1.1), or common ontologies (Icarte et al., 2017), in other cases researchers have tested the possibility for a machine to learn about inferences on the basis of just textual data (Rashkin et al., 2018; Trichelair et al., 2019).

1.3 Text-based Representations

While language has a lot to do with cognitive ability and mental processes, which are extremely difficult to retrieve and use as data, the output of communication efforts (i.e., texts and documents) is actually easy to retrieve and manipulate. Because of this, textual data has historically been used as the main source of information about language and linguistic phenomena for actual applications, excluding inner processes.
Over the years many models have been proposed for text-based representation (Naseem et al., 2021), from classical feature-based models to more recent low-dimensional distributional models (see Section 2.3). The starting point of these models is generally to simplify texts as *bag-of-words* (BOW), a list of words, disregarding underlying structures such as syntactic and semantic dependencies. While earlier BOW models also disregarded word order, more recent systems actually implement such information in their representations.
In particular, one of the main trends of state-of-the-art NLP systems is the committment on using Language Models (LMs), such as BERT (Devlin et al., 2019), trained using a neural architecture as a means to represent natural language sentences.
While Section 2.3 gives more details on how neural LMs work, the main take-away of such representations is that it can be learned directly from raw texts, on the basis of the statistical distribution of words in the corpus (Firth, 1957; Harris, 1954). In particular, LMs approach this core idea by training a neural network to predict the next word in a sentence, and the probability of a given sequence of

words (Bender et al., 2020).
Recently, the high rate of success of large LMs on certain tasks, such as question answering, sentiment analysis, question classification an sentence entailment (Qiu et al., 2020a) has caused a rise in the interest on them (Bender et al., 2020), in industry and academy alike. In particular, the hype caused byLMs regards the possibility for such systems to understand, comprehend and act on the basis of natural-language utterances in a human-like way once they are trained on a large enough corpus.
This hype is further fueled by efforts made in the analysis of said LMs through the use of probing tasks (Adi et al., 2017; Warstadt et al., 2019; L. Yu et al., 2020). These efforts have shown that LMs implicitly acquire some information about unspecified language-specific phenomena such as subject-verb agreement (Goldberg, 2019; Jawahar et al., 2019), semantic structures (Tenney et al., 2019), semantic similarity (Mikolov, Yih, et al., 2013; Rubenstein et al., 1965, and relational knowledge (Petroni et al., 2019; Safavi et al., 2021).

1.3.1 Issues with text-based learning

Despite the successes with recent implementations of LMs, and text-based representations in general, the hype behind them should be generally checked by looking more closely at how, and why, they work (Bender et al., 2020).
While such doubts on the inner workings of LMs have started to become more popular in recent years, in an early paper by Jones (2004) some of the issues which would become prevalent in recent years are already defined.
One such issue is the importance of data, in that the amount of data used is directly correlated to the performance of said LMs, with more data bringing about more powerful models. Nevertheless, the author highlights that, being based on statistical information, no amount of data would make the models match every possible situation.
Limitations in terms of statistics have also been recently highlighted in the more general field of Artificial Intelligence (AI) (Landgrebe et al., 2019, 2021), in particular in relation to the impossibility of developing an AI capable of solving complex systems (Landgrebe et al., 2022), understood as systems in which the components countinously interact with each others and with the system itself (Ladyman et al., 2013).
Since, by definition, natural languages are complex systems on their own, this

would mean that it is impossible to fully automate the processing of natural language, which does not mean that working on such systems is useless, but only that the scope should be narrowed down.

One further issue arises when investigating said models, since they inherit the opaqueness of the neural models they are based on. Because of the so called black-box nature of neural models (Dayhoff et al., 2001), the investigation of LMs is extremely complex and can take many different forms. One popular method, for instance, is to investigate which linguistic information is captured by neural networks (Belinkov et al., 2020), by employing specific probing classifiers (Alt et al., 2020) or by analysing neuron activation (Alt et al., 2020) and attention heads (Alt et al., 2020).

Another line of researches test how these models fare under pressure (Tenney et al., 2020), such as under adversarial settings (Belinkov et al., 2017; Ebrahimi et al., 2018; Jia et al., 2017; Papernot et al., 2016).

The key takeaway of investigating LMs is that gaps still exist between meaning and its representation, and that these gaps are of difficult analysis in a purely distributional environment, even in recent Transformer models.

For instance, Yanaka et al. (2019) investigate whether neural NLI models implicitly learn monotonic reasoning. The bad results show that such information is not encoded in any way by the models, which affects their generalization capabilities. The authors also show the integration of external data on top of text-only information greatly improves the results.

This is similar to what has been shown in the experiments on NLI by McCoy et al. (2019). The authors prove that neural models learn heuristics, rather than actual logical inference, to solve the final task. The authors also underline the need to inject knowledge coming from outside the text.

In Forbes et al. (2019), the authors investigate the extent to which neural LMs demonstrate physical commonsense reasoning. The idea is that humans possess an inherent knowledge of the physics of the objects present in the world, in particular regarding their physical properties. This knowledge makes it possible to reason about if-then inferences, which is something that is not found in LMs representation, except for those inferences that are explicitly written down in training texts.

While Petroni et al. (2019) have shown that LMs inherently learn some sort of relational knowledge, the experiments by Pörner et al. (2019) show that this retrieval of relational knowledge is actually based on entity names, rather than actual factual knowledge. This kind of factual knowledge, the authors highlight, has to be integrated in LMs from an external knowledge source.

Similar issues have also been found in relation to temporal knowledge, as seen in B. Zhou et al. (2020).

Finally, Niven et al. (2019) observe that the high performance of LMs on argumentation mining (Mochales et al., 2011) is essentialy derived from the exploitation of specific spurious statistical clues, rather than on actual information about arguments and semantics.

It is easy to see that there is a common theme in these works: since purely-textual LMs are based on using unstructured texts as the sole background knowledge for natural language representation, issues arise when information are not actually present in texts. This is true not only for information on a linguistic level (such is the case for dependencies and semantic arguments), but also for common-sense knowledge (such as world knowledge).
As previously mentioned, human communication is based on certain pragmatics principles (Grice, 1989), and one of the most important aspect of such principles is that not everything is said in conversations. Since this is to be expected, human speakers are also expected to act based on what is not said in the actual conversation, based on the background knowledge of the world derived from one's own life experience and information about the context as well.
But if a model has only the text as its source of knowledge, it would never be able to retrieve such informaion in a complete way. Some examples of such issues in relation to common-sense reasoning are found in Bouraoui et al. (2019).
Bender et al. (2020) maintain that meaning cannot be learned from form (i.e., texts) alone, and as such grounding to the real world is needed to fully *understand* natural language. The question then arises of what is the best way to integrate grounding into LMs. One option to integrate such ground to world knowledge is the training of the models on structured objectives (e.g. relation extraction or NLI) which would lead the system to learn *something*, but these systems have already been proven to still learn on the basis of simple heuristics.
More practically, while an extremely complex system such as GPT-3 (T. Brown et al., 2020) or BERT (Devlin et al., 2019) would be able to fool a human interlocutor by making use of their extremely complex statistical knowledge of linguistic forms' distribution, without an actual mapping of language to real-world entities this could not cover all the possible linguistic examples which are, instead, possible to be covered and understood by humans.

Given these bases, it is clear that texts on their own are not enough to learn models for natural language representations, but that external, grounded information is needed in order to solve many of the present issues.

1.4 Graphs for Knowledge Representation

Since distributional representations based on only texts are shown to have inherent issues, the integration of structured and explicit external information has been proposed as a way to solve unsolved problems. One of the possible sources of such external information is represented by knowledge graphs (KGs), and graph-based knowledge representations in general.
In recent years, in fact, KGs have become one of the main sources of data for the task of integrating structured knowledge into distributional representations for natural language (see Section 2.4), which can help with the above-mentioned need of integrating structured, non-textual information together with text-based representations.

The use of graph-based data representation has seen a rapid growth, both in terms of development of graph-specific techniques and in terms of applications of such structures (Q. Wang et al., 2017). In NLP, in particular, large Knowledge Graphs such as YAGO, DBpedia, Freebase and Wikidata have been implemented in a variety of tasks, such as semantic parsing (Berant et al., 2013; Heck et al., 2013), named entity disambiguation (Hakimov et al., 2012; Zheng et al., 2012) and question answering (Bordes, Chopra, et al., 2014; Bordes, Weston, et al., 2014).

Motivation

Ever since the announcement of the Google Knowledge Graph (Singhal, 2012), the development of graph-based structures for storing, maintaining and accessing knowledge has gained a lot of attention, as proven by the implementation of such graphs by several companies (Hogan et al., 2021) following Google's example, and the attention gained in relation to several topics of research (Ji et al., 2021).
Despite this recent surge in interest, the idea of using graph-like structures to represent data has a fairly long history, with the use of a network-based representation of concepts being proposed by the Greek philosopher Porphyry (Sowa, 1999), and with the term "knowledge graph" appearing as early as 1972 (Schneider et al., 1973), as a means to store information used in an instructional system.
In the area of natural language representation, the use of a network-based structure to represent utterances was first proposed by R. Quillian (1968) and then later developed by Allen (1987) in the structure that then came to be known as *semantic*

network.
Over the years, the line between knowledge graphs and semantic networks began to blur, and, following L. Zhang (2002), knowledge graphs are generally understood as a more-specialised form of semantic networks.
As stated by Angles et al. (2008), the use of graph-based data modeling is particularly beneficial in those areas where *information about data interconnectivity is as important as the data itself*. This definition applies to many areas, including natural language, since one of the main axioms of natural language is that words and utterances generally cannot be understood on their own, and that their meaning arise from their interconnection to other elements of the context they are used, and how the elements are connected among each other. Broadly speaking, this is also the core idea behind the distributional hypothesis (see Section 2.1.2) and compositionality (Bender et al., 2015).
In such cases, the use of graphs to represent information offers several advantages:

- the modeling of data using graphs is symbolic rather than distributional (i.e., it is easily readable by human users);
- queries over graphs can make use of specific graph operations and algorithms for advanced applications (e.g., finding shortest paths, finding subgraphs, or using specific metrics such as the centralness of nodes, etc.);
- graphs allow the data to be developed in a flexible way, by giving the possibility of postponing schema definition, which is especially useful for the representation of incomplete knowledge, as shown by Abiteboul (1997).;
- graphs allow for the implementation of many standard knowledge representation formalisms (e.g., ontologies) to define the semantics of the structure and improve access and reusability to data,
- in recent years, the development of machine learning techniques specifically tailored for graphs, particularly in the form of Knowledge Graph Embeddings, has also increased the tools available when working with such data structure (Q. Wang et al., 2017).

Given these benefits, knowledge graphs have become a popular means to share knowledge on the Web, particularly for large amounts of interconnected data. In this regard, Freebase (Bollacker et al., 2007), Wikidata (Vrandečić et al., 2014), DBpedia (Lehmann et al., 2015) and Yago (Hoffart et al., 2011) represent well-known examples of multi-domain KGs, which have been amply used as a source of reliable and structured information.
Graphs might also be domain-specific, and they offer an important source of information for tasks related to more fine-grained knowledge representation. Graphs have been employed to represent domains such as Government Data (Holm et al.,

2012), tourism (Maturana et al., 2018), Life Science (Callahan et al., 2013) and more (Hogan et al., 2021).
This is true for linguistics as well, with several language resources being published using graph-based representations. For instance, in frame semantics (Fillmore, 1977) words are understood as being connected to specific *situations*, along with all the essential knowledge for the situation to be understood.
The verb *to sell*, for instance, evokes (i.e., it refers to) the situation of commercial transfer, which involves (at least) a seller, a buyer, goods and money.
These situations are represented as *frames* connected to their arguments using a network of nodes and edges. By extension, FrameNet[14] (C. F. Baker et al., 1998), the linguistic resource based on this theory, makes use of a graph-based representation to share data about frames.
Similarly, in the lexical resource WordNet[15] (Miller, 1995), synonymous words are linked together into *synsets*, which are then connected among each other according to their semantic relations (e.g., synonymy and meronymy), creating nets of concepts.
This data can be incredibly useful in solving many of the current issues we are facing in NLU, since it explicitly shows the connection that distributional models try to implicitly predicts, and thus in recent years efforts have been made in order to implement it, together with contextual information, in a coherent and meaningful way. Section 2.4 describes some of these efforts.
While this is true for knowledge graphs referring to linguistics specifically, domain-specific knowledge graphs can be of extreme benefit as well, in particular for those cases in which raw text for a specific domain is hard to find, making it difficult to train a distributional model.

1.4.1 Models for Graph representations

In terms of practical applications, the two main frameworks for graph data modeling are represented by the Resource Description Framework (RDF) and Labeled Property Graphs (LPG). This section aims at giving a general overview of both models, while also describing benefits and drawbacks for each one.

[14] framenet.ics.berkeley.edu/fndrupal/
[15] wordnet.princeton.edu

Linked Open Data and Resource Description Framework

RDF represents the go-to data model used to share data on the Web in the context of the Linked Open Data paradigm (Bauer et al., 2011). Since this is the case, the main method through which this data frame is published is in a structured textual form which follows the RDF language, standardized by the World Wide Web Consortium[16] (Manola et al., 2004).

More practically, each RDF graph is composed of information represented by triples of type `subject node, relation, object node`. Three possible nodes exist: URIs, literals and blank nodes. URIs are string sequences used to identify resources on the Web, such as `www.wikidata.org/wiki/Q12418`, which is the URI identifier for the entity representing the Mona Lisa on Wikidata.

Literals, on the other hand, represent a value from a certain data type, for instance a string, a date, or a numeral. Blank nodes represent nodes with no identifier, and can be used to represent variables.

In a triple, relations are generally identified by their URI as well. In order to give a more compact and cleaner view of the graph, prefixes can be defined so that URIs can be abbreviated. For instance, we can define a prefix `wd` for www.wikidata.org/wiki/ so that the above-mentioned URI defining the mona lisa can be abbreviated as `wd:Q12418`. A triple is such as follows:

```
@prefix wd:  <http://wikidata.org/wiki>
@prefix wdt:  <http://wikidata.org/wiki/Property>
wd:Q12418  wdt:P276  wd:Q19675
```

This triple describes the existence of a relation `P276` (*location*) between the entity `Q12418` (*Mona Lisa*) and the entity `Q19675` (*Louvre Museum*). An RDF graph is composed on many such triples.

The strongest benefit of using RDF is represented by the use of permanent URIs to map nodes and relations to a unique identifier, which can be shared across several graphs, thus improving reusability of data.

Furthermore, as mentioned already, RDF is highly standardized, and its usage has a long history in the literature (Ali et al., 2021). This also means that many knowledge bases on the Web are freely available in RDF format to be accessed[17], despite the current state of RDF archiving being still underdeveloped (di Buono et al., 2022; Pelgrin et al., 2021).

The access to this data is generally done through a specific language called SPARQL,

[16] https://www.w3.org/RDF/

[17] https://www.w3.org/wiki/DataSetRDFDumps

also standardized by the World Wide Web Consortium[18]. The use of a standardized query language makes it so that querying data generally follows the same rules regardless of the specific dataset that is being accessed. SPARQL also allows for specific optimization techniques (Rabbani et al., 2021; Saleem et al., 2018), and it has been widely used as the desired output for general Semantic Parsing models (Cimiano et al., 2014; Luz, 2019; Luz et al., 2018, which represent the main topic of Section 2.2.

Furthermore, since one of the main principles of this framework is to make data linked *in a coherent way*, many ontologies (both for general domain applications and domain specific) have been proposed through the years to describe data in a coherent way. These ontologies are generally freely available on the web[19], and they follow the Web Ontology Language[20] (OWL) language to represent knowledge.

In the field of linguistics and the development of language resources, one of such ontologies is represented by OntoLex-Lemon[21] (J. P. McCrae et al., 2017), which has been used extensively in several resources (S. W. Brown et al., 2017; Gangemi et al., 2016; Rospocher et al., 2019).

Similarly, in Chiarcos et al. (2017), the authors propose the development of CoNLL-RDF, a rendering of the CoNLL tsv format into RDF. The main objective of such effort is to create a vocabulary that makes use of the extensive history and range of applications of the CoNLL format, while facilitating the integration of Linked Open Data (LOD) principles[22]. Furthermore, mapping from the CoNLL tsv format to its RDF counterpart is pretty easy, with resources being developed to do so freely available[23] (Chiarcos et al., 2017; Chiarcos et al., 2020; Chiarcos et al., 2021).

While the RDF framework offers many benefits, in particular regarding the reusability of data and a wide range of available tools, resources and schema to develop one's own graph, many drawbacks are still present in this regard. First of all, as mentioned already, the presence of a high amount of ontologies, schema and datasets means that some overlapping of data is present, while interoperability of data is seldom applied (di Buono et al., 2022; Ding et al., 2006), which means there is still a strong need to bridge data and metadata between datasets and repos-

[18] https://www.w3.org/TR/sparql11-overview/

[19] https://www.w3.org/wiki/Lists$_{o}$f$_{o}$ntologies

[20] https://www.w3.org/OWL/

[21] https://www.w3.org/2016/05/ontolex/

[22] https://www.w3.org/wiki/LinkedData

[23] https://github.com/acoli-repo/conll-rdf

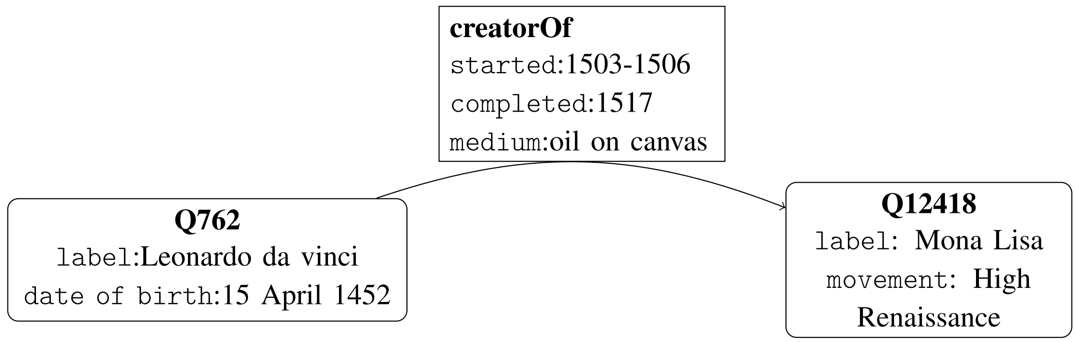


Figure 1: Example of a labeled property graph

itories. This is made particularly difficult by the presence of hundreds of ontologies, which would need to be *aligned* before any bridging can happen.

Labeled Property Graphs

While RDF is a schema primarily designed to describe and share resources on the Web, the Labelled Property Graph (LBG) is a database model, designed to specifically manage and work with graph-like data, as opposed to relational databases (Angles, 2018). The main feature of such a graph database model is that it can encode, for each node or edge, a (possibly empty) set of properties (Angles et al., 2019).

This leads to differences in how data is encoded. For instance, given a triple (`LeonardoDaVinci`, `creatorOf`, `MonaLisa`), additional information could be added regarding the date of completion of the work, or specifics about the medium, by adding specific *properties* to the relation `creatorOf`, as shown in Figure 1.

The main use of such properties is to define information about nodes that would not connect any two nodes. For instance, RDF literals might be represented as properties rather than triples without losing any structural integrity of the graph. On the other hand, removing these relations greatly improves traversals' performance at query time.

In RDF, the only way to encode additional information about *relations* rather than nodes is through *reification*, that is the process through which a triple is encoded as a node of its own, with subject, predicate and object being its attributes, so that meta-information can be added (Hernández et al., 2015). The process of reification is graphically shown in Figure 2.

Hernández et al. (2015) go into details about the types of reification available, but the core idea stays the same. This process is non-trivial, and the absence of a clear-defined semantics and formal description in the RDF Primer makes its use particularly difficult (V. Nguyen et al., 2014).

$$\text{wd:Q12418} \xrightarrow{\text{wdt:P276}} \text{wd:Q19675}$$

statement — hasSubject ⟶ wd:Q12418
statement — hasproperty → wdt:P276
statement — hasObject ⟶ wd:Q19675

Figure 2: Example of reification: the graph on the left represents a basic RDF triple, the one of the right its reified representation.

One additional feature of property graph is that it does not need a predefined schema to be implemented. Actually, the schema of such a structure is dynamically built as the graph grows: this makes the framework especially useful in case of dynamic data (language falls into this category). Furthermore, property graph environments are generally designed with optimization in mind: in particular, this means that specific graph algorithms, such as shortest path finding and traversing techniques, are already implemented and quick to use (for instance in path analysis and deep queries). For this reason, this kind of graphs are generally used in industry as opposed to the RDF schema (Angles, 2018).
On the other hand, reusability of data becomes an issue when using such data. First of all, while RDF is a schema and is usable through simple text files, property graphs generally require a specifically designed framework, such as Neo4J[24] or the Python library NetworkX[25], each with its own tools and peculiarities. This makes it hard for graphs to be reused unless they are designed using the same framework.
The presence of multiple frameworks and the lack of a pre-defined schema also mean that, while there are some general best practices on how to implement property graphs[26], no one commonly accepted set of syntax and semantics is defined

[24] https://neo4j.com/
[25] https://networkx.org/
[26] https://neo4j.com/developer/guide-data-modeling/,https://docs.microsoft.com/en-us/graph/best-practices-concept,https://www.dataversity.net/graph-databases-best-practices-and-new-developments/

in the community, which means that data from different sources will rarely be coherent, with a negative impact on interoperability and reusability.
Finally, while RDF has its own query language in SPARQL, which allows users to access many RDF graphs, property graphs lack a common one, which directly affects the accessibility to online databases. A publication called the GQL manifesto[27], proposes the standardization of such graph query language through the fusion of the three most used query languages for graphs, but at the time of writing this has yet to become a reality.

1.5 Research Question

Given the presented topics and issues in relation to the task of representing natural language for NLU tasks and applications, this section describes the main research questions which are at the core of the efforts described in the course of this work. The first question is concerned with a general analysis of whether the integration of external knowledge together with text-based representation in order to improve the final results of NLU models. In particular, the focus of such a question is on two main tasks: semantic enrichment and relation extraction.

> *(RQ1) What are the effects of graph-based knowledge integration on NLU-related tasks of semantic enrichment and relations extraction?*

While this question covers several of the theoretical issues previously defined in relation with semantic theories and natural language representation, it is also of interest whether there is actually an improvement in the final results through the integration of explicit, structured information represented using graphs.
This kind of integration can take many forms, since graph-based knowledge can come in many sizes and shapes, and the second research question deals with this topic.

> *(RQ2) Which is the best format for graph-based knowledge integration?*

[27]https://gql.today/

More specifically, this work is interested in investigating the use of knowledge graphs, which can generally come as either RDF or labelled property graphs. These paradigms are integrated in different NLU applications, and investigated accordingly, particularly in relation to the task of question answering.

1.6 Contributions

The main contribution of this thesis is represented by a series of efforts made in relation to graph-based knowledge integration for natural language understanding. In particular, they refer to three main lines of work:

- The definition of a language-agnostic methodology for the integration of graph-based knowledge (in the form of ontologies) for concept retrieval and semantic enrichment (di Buono et al., 2023; Nolano et al., 2022a, 2022b);
- a preliminary analysis of graph-based models for relation extraction models against distributional representations, in a standard environment and in an adversarial evaluation framework;
- the definition of methodologies for the development of a QA system based on open-data knowledge (Nolano et al., 2021).

1.7 Dissertation Outline

This dissertation is organized as follows:

- Chapter 1 introduces the general topics discussed in this thesis and provides an outline of the full work. In particular, the chapter focuses on issues in relation to the understanding and representation of natural language, both in relation to linguistic levels of analysis and theoretical foundation, and in terms of practical implementation in real-world systems.
- Chapter 2 offers a literature review on the broad topic of representation in Natural Language Understanding, according to two main frameworks: formal and distributional representations. The chapter also gives an overview of methodologies implemented in order to represent natural language according to these two frameworks, namely through semantic parsing and neural word embeddings. A portion of the chapter is further dedicated to

representations that try to implement both formal and distributional frameworks together in a single environment.

- Chapter 3 proposes a methodology for the extraction of concepts and semantic enrichment of texts, based on ontological information. In particular, the chapter focuses on the description of two different methods implemented for domain-specific semantic enrichment, for the domain of archaeology and cultural heritage in general.
- Chapter 4 investigates the graph-based knowledge representation and injection for the task of relation extraction. In this chapter textual models based on neural networks are fared against graph-based representations making use of both textual and structural information from a background knowledge base, both in a standard and in an adversarial environment.
- Chapter 5 describes efforts made in relation to the development of Knowledge-graph Question Answering over Linked Open Data. The chapter describes two efforts in this regard, both making use of graph-based representations of questions to query over a background knowledge graph.
- Chapter 6 draws conclusions and provides an outlook for future work.

Chapter 2

Natural Language Representation

This chapter gives a general overview of related works with regard to the general topic of natural language representation. In particular, the first section gives an overview of two main frameworks of representation, namely formal and distributional representations. The second section describes the task of semantic parsing, that is the explicit mapping of sentences to machine-understandable representations. The third section focuses on the implementation of neural networks to derive representations from texts for NLP purposes, generally seen as the go-to way to represent natural language utterances. Finally, the fourth section defines efforts made in relation to the use of graph-based structures to represent natural language.

2.1 Methods of representation

One of the main topics of discussion in NLU is what *form of representation* should be used to map semantic information from a given natural language utterance to a machine-readable format (Clark et al., 2007). While in recent years this issue is becoming more and more a practical problem concerning which one is the better method to use given a specific task, the foundational ideas on the topic have their roots in linguistic, philosophical and logical theories (Pirozelli et al., 2022).

The choice of which representation method to use when developing a system for NLU is thus of critical importance, both in terms of the theoretical background

being followed and in terms of the final results (Bender et al., 2020). In particular, the most commonly used forms of semantic representation usually fall into one of two frameworks: formal representations and distributional representations.
These two frameworks differ both in terms of their foundational ideas and in the way they are actually implemented in real-world scenarios. Following the definition proposed by Emerson (2020) and Koller (2016), the two paradigms can be defined in terms of top-down and bottom-up approaches: while the former are interested in reaching a specific objective, the latter is more interested in expanding models based on a given theory or tool.
More specifically, formal semantics generally follows a top-down approach (with the specific objective being the mapping of natural language to a logic, symbolic format), while distributional semantics can be defined as a bottom-up approach to the same natural language representation problem (i.e., it is based on distributional representations of language and it expands models based on this representation).
This difference also has an impact on how the paradigms are viewed in the current state of the art. In fact, since formal semantics generally has as a goal the principles of truth-conditional semantics and entailments, as previously defined in Section 1.1, and since these well-defined goals have yet to be reached, the successes of formal semantics are generally undervalued in the current state of the art, while, on the other hand, distributional semantics generally seemingly moves from successes to successes, since no clear goal is set as long as distributional representations are used (Koller, 2016). Because of this, in recent years it has become extremely common to look at the successes of distributional semantics approach without taking into account its intrinsic issues (Bender et al., 2020).
On a more practical level, another important difference between the two frameworks is in their application scope: while formal semantics is generally interested in the representation of larger units (e.g., sentences and utterances), distributional semantics is more interested in the representation of single tokens (Venhuizen et al., 2022).
The next sections give a brief overview of the two methodologies thus introduced, together with more specifics in terms of foundational ideas and implementation techniques.

2.1.1 Formal Representations

Historically, the earliest attempts of representing meaning in some sort of unambiguous and general form were based on the usage of a formal and logical language (Wintner, 2002). The main benefit of using such logical languages is that they are, by definition, unambiguous, as opposed to natural languages, and as such can be implemented together with automatic systems (Allen, 1987).

The idea of using a formal metalanguage to represent the meaning of a natural language utterance as its root in works of linguists such as Frege (see Section 1.1), and, in particular, Montague (Montague, 1970a), who set the basis for what came to be known as *Montague semantics*.
The core principle of this theory is that it is possible to represent natural language using logic languages, and that the meaning of a sentence can be understood as a *function* of the meanings of its parts and the way syntax combines them, just as an expression is a formula of its part. Furthermore, it enforces the idea that a well-defined metalanguage is an essential part in representing meaning (Partee, 1981).

The use of a metalanguage to describe the semantics of utterances has found a lot of popularity in NLP since the work described by Greenberg (1949), which made use of first-order predicate calculus to describe semantic components.
Given this idea, over the years many formal languages have been proposed for the task of representing meaning from natural language utterances. Two main families of formalism have been defined and are commonly used: **logic based formalism** with its root in philosophy and mathematical logic and based on logic languages such as first order logic; and **structured formalism** which is usually linguistically or psychologically motivated, and based on structures such as graphs and frames.

Logic based formalisms

One of the most flexible and commonly used logic based representations for language is First-Order Logic (FOL).
FOL makes use of functions and constants to express meaning, while also allowing for variables and quantifiers, all extremely useful features for the representation of natural language.
For instance, a question such as *Who painted the Mona Lisa?* would be represented using FOL as:

(1) $\exists y.hasName(x,y) \land creator(MonaLisa,x)$

Despite being recently used in QA as described by P. Liang (2016), and despite all its flexibility and expressivity, FOL shows some heavy limitations.

For instance, FOL does not allow for the aggregation and manipulation of sets of entities. Thus, questions such as *How many authors worked on the Sis-*

tine Chapel? are impossible to be represented in FOL, its logical interpretation asks for the aggregation and subsequent counting of all the x that participated in $creator(SistineChapel, x)$.

One possible solution to this issue is the augmentation of FOL with lambda calculus (LC), as proposed by Carpenter (1997).
LC is based on the use of λ-terms, which group together all the possible values of a variable, given certain conditions. Thus, the question *How many authors worked on the Sistine Chapel?* can be represented as

(2) $count(\lambda x.creator(SistineChapel, x))$

With λx grouping together all the entities that can be found in a relation `creator` with `SistineChapel`.
LC has been used in recent works in NLU (Kamath et al., 2018), with applications in the creation of queries for DBs (L. S. Zettlemoyer et al., 2005); interaction with conversational agents (Artzi et al., 2011); and instructions mapping for robot interaction (Artzi et al., 2013).

P. Liang et al. (2013) propose a more compact version of LC, *Lambda Dependency based Compositional Semantics* (λ-DCS), where variables are eliminated and existential quantifications are made implicit. Thus, the aforementioned question would be represented as:

(3) $count(creator.SistineChapel)$

Applications of λ-DCS for natural language representation can be found in several works, such as Berant et al. (2013), Krishnamurthy et al. (2017), and Pasupat et al. (2015).

Structured formalism

Another possible paradigm is represented by **structured formalisms**, which are based on certain data structures rather than logical formulas. The most popular of such data structures for NLP purposes are graphs and frames.

Graphs have seen many implementations for NLP, such as in the form of **Semantic Networks**, first proposed by R. Quillian (1968) and later developed for linguistic purposes by Allen (1987). In this paradigm, the graphs used to represent meaning represent objects in the world in the form of nodes, and arcs representing relationships between these objects, as described by a source of knowledge.
In comparison with logic representations, using graphs offers several advantages:

1. they are generally easier to read for humans;
2. graph algorithms can be leveraged to implement better reasoning;
3. graphs following LOD principles (see Section 1.4) are easy to share and integrate with each other to increase the amount of information covered.

In recent years, similar representations have been used in the form of *semantic queries*, which can be implemented to retrieve the correct answer for natural language questions, given a background database. This is the case, for instance, of the work described by Yih et al. (2015), in which a query graph is automatically built from an input question. The model was later implemented and improved by Bao et al. (2016) and Sorokin et al. (2018).
One of the main issues regarding this kind of representation is that, since they are based on an external ontology for representation, the reuse of data is heavily dependent on the implementation of a given task- and domain-specific definition (Gracia et al., 2012).

Another kind of structured representations is the frame paradigm, based on the work by Fillmore (1968), in particular his work on Case grammar.
According to this paradigm, each situation puts certain constraints on which cases (i.e., semantic roles) it needs in order for a piece of communication to have a complete meaning. These constraints are reflected in both the syntax, in terms of which syntactic units appear in a given sentence, and on semantics, since a certain number of obligatory semantic arguments are needed for an utterance to make sense, each appearing in certain syntactic roles.
For instance, the verb *to paint* requires at least two elements to make sense: a human painter and an object being painted. Given this assumption, the sentences *Leonardo da Vinci painted the Mona Lisa* and *The Mona Lisa was painted by Leonardo da Vinci*, despite the different structure, represent the exact same situation, in which *Leonardo da Vinci* is always the agent of the action, since the verb requires an human agent, and the *Mona Lisa* is always the object painted.
Furthermore, this also underlines the presence of a strong link between syntax and the underlying meaning of every utterance (M. Baker, 1988; Gruber, 1965; Jackendoff, 1972, 1990), which nevertheless does not mean that the connection between the two is trivially retrievable (Levin, 1993).
The deep connection between surface forms and semantics gives theoretical background to the mapping of an utterance to its representation using frames. These concepts later converged in the frame paradigm, which has its roots in AI (Huesmann, 1974; Minsky, 1974) where frames (or scripts, as they were referred to in the earlier stages) are used to describe procedural or stereotyped knowledge.
In linguistics frames were mainly adopted by Fillmore in what came to be known

as frame semantics (Fillmore, 1977), in which case grammar is extended to link semantics to encyclopedic knowledge. According to this theory words can be linked to a certain frame which would describe their meanings in terms of connections to other linguistic entities.
For instance, the above-mentioned verb *to paint* generally describes a situation in which a *representation* is being *created* by a *creator*. This frame is *evoked* by the surface form of the verb *to paint*, but in general it is more correct to say that it is related to a certain situation of *creating a physical artwork*, which is also evoked by certain nouns such as *Mona Lisa* and *paintings*.

While there is no general agreement on which and how many roles are needed for this representation, a certain level of consensus exists according to specific arguments, such as the agent (i.e., the *doer* of the action depicted by the predicate) and the patient (i.e., the *undergoer* of the action depicted by the predicate). Furthermore, these arguments are generally *indexed* so that they can easily be mapped to the underlying syntax of the sentence (Dowty, 1991): because of this, a generally accepted practice is to assign the label `arg0` to the agent, the label `arg1` to the patient, and so on for other optional arguments.

Frame-based representation has been used in the development of several annotated resources. Among the most popular ones there are FrameNet[1] (C. F. Baker et al., 1998), a resource which is directly based on Fillmore's frame theory, and Propbank[2] (Palmer et al., 2005a).
One of the main differences between the two resources is how they represent predicate's arguments. On one hand, FrameNet implements frame-specific annotations, which means that the frame *painting* has labels such as `creator` and `representation`, while the frame *examination* has labels such as `examination` and `examinee`. PropBank, on the other hand, makes use of a predicate-agnostic indexing (thus leaving `arg0`, `arg1`, etc. as labels). The two different annotations of the same sentence are thus as follows:

- **FrameNet**: [Leonardo da Vinci]$_{\text{creator}}$ [painted] [the Mona Lisa]$_{\text{representation}}$
- **PropBank**: [Leonardo da Vinci]$_{\text{arg0}}$ painted [the Mona Lisa]$_{\text{arg1}}$

While FrameNet is specifically designed to provide more fine-grained semantic information about frames, PropBank is more focused, by design, on providing a data for training statistical systems (Palmer et al., 2005a), in which case the simpler annotation schema is a benefit, since it reduces the possibility of overtuning the trained system.

[1] https://framenet.icsi.berkeley.edu/fndrupal/
[2] https://propbank.github.io/

Furthermore, while FrameNet's representations are focused on the semantic of a sentence, PropBank's representations have a stronger connection to its syntax. One of main effects of this difference can be seen when dealing with verbs evoking the same frames but with different syntactic structures, as explained by the following examples, proposed by Palmer et al. (2005a):

- **FrameNet**:
  - [Chuck]$_{\text{buyer}}$ [bought] [a car]$_{\text{goods}}$ [from Jerry]$_{\text{seller}}$
  - [Jerry]$_{\text{seller}}$ [sold] [a car]$_{\text{goods}}$ [to Chuck]$_{\text{buyer}}$
- **PropBank**:
  - [Chuck]$_{\text{arg0}}$ [bought] [a car]$_{\text{arg1}}$ [from Jerry]$_{\text{arg2}}$
  - [Jerry]$_{\text{arg0}}$ [sold] [a car]$_{\text{arg1}}$ [to Chuck]$_{\text{arg2}}$

This means that, for FrameNet annotations, the roles for *Chuck*, *Jerry* and *car* are always the same, regardless of the realization of the specific frame. In PropBank, on the other hand, the roles are more strongly connected with the underlying syntax of the sentence, and thus change as the sentence changes.
Finally, while FrameNet includes nouns and adjectives as well as possible frames' realization, PropBank addresses only verbs as anchors for frames' realization.

The main benefit of using frame structures in a model is that the constraints inherently posed by their semantic meaning and their representations might be leveraged to have a more robust system. For instance, in a sentence such as *I saw him playing the bass*, the word *bass* might be ambiguous for a system, which could understand it as either an *instrument* or a *fish*. In this case, knowing that the `performing_arts` frame is evoked by the verb *to play*, would lead the system to understand *bass* as the `medium` used to play music with, and thus disambiguate it. On one hand, this kind of knowledge is of extreme importance in natural language processing, but, on the other hand, the lack of availability for frame resources for certain languages, such as Italian, makes it difficult to implement in every scenario.

2.1.2 Distributional Representations

The theoretical roots of **distributional representations** are found in the linguistic theory of distributional hypothesis, which was laid out by authors such as Firth (1957) and Harris (1954). The core idea behind this theory is that the meaning of words (along with every other linguistic aspect) can be inferred from the distribution of linguistic entities given a large enough number of examples of that language, that is by the word's context of use. By extension, this means that se-

mantically similar words will share similar contexts, while semantically dissimilar words will have little to no context in common.
Distribution, in particular, is defined as the frequency in which certain linguistic entities occur among each other, with the conclusion being that linguistic entities occurring in similar contexts will share similar linguistic features, including (but not limited to) meaning.

The main contribution of this theory is that there is no need for any non-linguistic information to represent the meaning of natural language utterances. That is to say, given a large enough source of textual information and a way to represent differences among linguistic entities in terms of occurrences, these differences can be used to infer information about words and sentences.
While this theory was introduced as being implementable at any linguistic level, in particular in the works of Harris (1954), in recent years it has mainly been applied in relation to words and sub-word entities.
Representation methods based on this hypothesis represent words from a given language/vocabulary in relation to their distribution given a large enough context of use (e.g., a corpus). Since the advent of modern computing, the large amount of textual data freely available on the Web, and the increasing power of hardware, made it so that this representation has lately become the default choice for natural language representation (Ferrone et al., 2020).

Vector Semantics & Word Embeddings

The standard method for distributional representation is to represent each word by mapping it to a *vector*, that is a list of numerical values that represents specific linguistic information. The main benefit of using vectors is that they allow for the integration of many algebraic techniques, such as cosine or Euclidean distance to compute semantic similarity between two words.
One of the earliest applications of vector representation for natural language can be found in the work of Jones (1964), who propses the representation of word meaning by using boolean vectors based on information from an external thesaurus.
One of the most important steps regarding vector semantics was the introduction of the above-mentioned distributional hypothesis to compute vectors. In this sense, the original core element used to build such vectors is a *co-occurrence matrix*, where rows represent words, columns represent contexts (e.g. other words in the vocabulary, or documents), and cells contain the frequency of words given

a the desired context (e.g., how many times that specific word co-occur together with another word given a certain context window, or how many times it appears in a given document).
One of the main issues in using such a co-occurrence matrix for semantic representation is that raw frequency counts are rarely, if ever, informative of meanings. This is especially true since the distribution of words in language approximates to a Zipfian distribution (Fagan et al., 2010; Zipf, 1932). According to Zipf's law, in fact, the frequency of a word is inversely proportional to its statistical rank, which means that the most frequent words appear with an extremely high frequency, while least frequent words appear extremely rarely.
One of the main consequences of this property is that natural languages have a very few extremely frequent words, a small group of words that appear with an average frequency, and a large group of words that appear with a rare frequency[3]. Taking this into account, and the fact that generally the most frequent words in a language are function words, there is a need to adjust the statistical information about the data in order to represent meanings, so that more informative events get emphasized, while less informative events get ignored.

The first efforts made to fix these issues have been to *weight* the values in the matrix's cells, that is by integrating a specific function on top of the raw frequency counts. One of the earliest methods in this regard is the *term frequency x inverse document frequency* value (tf-idf) (Jones, 1972) to re-weight term-document matrices. This method has been applied, for instance, by Salton et al. (1988) for the task of automatic information retrieval.

Another popular method for re-weighting is *Positive Pointwise Mutual Information* (PPMI) (Niwa et al., 2003), which has been proven to perform well on term-term matrices (Bullinaria et al., 2007).

Issues are also found in relation to the optimization of such distributional models. First of all, the vectors created using these early models are generally sparse, in particular when dealing with term-term matrices. Most words, in fact, do not co-occur with every other word in the vocabulary, which leaves many cells with a value of 0 (Clausi et al., 1998).
Furthermore, using words to represent context in columns means that the length of the vectors is equal to the length of the vocabulary of the corpus, which creates extremely long vectors with a heavy effect on performances. This, combined with the large presence of zeroes, means that the results would be represented by ex-

[3]For instance, only 135 words are needed in order to recreate half of the 1-million-token Brown Corpus, as observed by Fagan et al. (2010)

tremely long vectors filled with empty values (Sahlgren, 2005).
Over the years, in order to solve both these issues, many techniques have been proposed to create shorter, denser vectors generally referred to as **word embeddings**. The task is thus to compress the amount of data present in the matrix, while also preserving information about the relations between words and contexts. Early on, this was done using linear algebraic methods for dimensionality reduction, such as Principal Component Analysis (PCA) (Hotelling, 1933; Pearson, 1901), Independent Component Analysis (ICA) (Lee, 1998) and Random Indexing (RI) (Kanerva et al., 2000; Kanerva, 1988).
One of the most popular methods is the use of Singular Value Decomposition (SVD) for matrix factorization. The application of this technique to a term-term matrix is called Latent Semantic Analysis (LSA) (Deerwester et al., 1990; Dumais et al., 1988). In linguistics, LSA has been used for several tasks mainly in relation to information retrieval, such as topic modeling (Kim et al., 2020; Valdez et al., 2018), document classification (Ju et al., 2015), synonyms retrieval (Ekštein et al., 2013), and extraction of semantic patterns (Altszyler et al., 2016).
Most interestingly, Landauer et al. (1997) analyze LSA from a cognitive point of view, by employing the model in order to explain some specific phenomena in human speech and understanding.

More recently, the pioneering work by Bengio et al. (2003) has proven the benefits of using neural models rather than algebraic or statistical processes for the implementation of word embeddings. In neural word embedding models, rather than first learning a co-occurrence matrix and then reducing its dimensionality, the embeddings are learned directly from a large corpus by teaching the model to solve a given task, usually a general one such as word prediction. In this case, the task is generally referred to as *language modeling*.
The more recent neural models generally reach state of the art results for many applications and datasets: named entity recognition (X. Wang et al., 2021); reading comprehension (K. Sun, Yu, et al., 2019); semantic parsing (J. Zhou et al., 2021) and more[4].
As mentioned already in the beginning of this section, this is to be expected: distributional representation has, at its core, a hypothesis rather than a specific goal (as it is the case for formal semantics), because of this, any step is considered a success, regardless of the direction and the inherent issues present in the models.

[4]A list of current state-of-the-art models for NLP can be found at http://nlpprogress.com/

2.2 Semantic Parsing

The task of mapping natural language utterances to representations of their meaning using an explicit formal language is generally defined as **semantic parsing** (Kate et al., 2010).

Over the years semantic parsing has represented a central topic of research in natural language understanding (NLU). The act of mapping a natural language utterance to a structured meaning representation has several benefits (Kamath et al., 2018), such as:

- it helps solving the ambiguity inherently present in all natural languages;
- it allows for reasoning and inference;
- it is extremely flexible, being re-usable in many applications.

Furthermore, as mentioned already, formal meaning representations resulting from semantic parsing are generally understandable and interpretable by humans as well as machines. This makes these kinds of models easier to maintain and update in contrast to modern distributional models.

Generally, models for semantic parsing differ along two main dimensions: the logical language used to represent meaning, and the techniques implemented to actually map the natural language utterance.

While the former has been discussed in Section 2.1, the techniques which can be implemented, usually fall into one of three categories: rule-based, statistics-based and neural, as it is the case for many other NLP tasks.

Another level of difference is described by Cheng et al. (2017), where the authors define two main approaches: under the first one, the mapping from utterances to meaning representations is done directly via the use of a task-specific grammar (Berant et al., 2013; Groschwitz et al., 2015; Kwiatkowksi et al., 2010; P. Liang et al., 2011; L. S. Zettlemoyer et al., 2005); in contrast, under the second one the utterance is first parsed to an intermediate, task-independent representation according to a syntactic parser, which is then mapped to a semantic representation (examples of this approach can be found in Cheng et al., 2017; Gardner et al., 2016; Kwiatkowski et al., 2013).

The main benefit of working under the latter is that the intermediate interpretations can be reused afterwards, even for different tasks, which would further improve the reusability of the final model and its parts. Nevertheless, as expressed in Bender et al. (2015), in recent years the lack of an explicit connection between surface linguistic representation and their underlying meanings (which is the core idea behind using self-supervised neural networks), represents an important drawback of recent proposed models for semantic parsing. Furthermore, the recent focus on domain-specific tasks rather than general-domain applications also hinders the aim of a general-purpose natural language understanding.

The following sections define some of the proposed models for semantic parsing.

2.2.1 Rule-based systems

Rule-based paradigm has mostly been used in early semantic parsing systems. There are several benefits of using rule-based systems: they do not require a lot of data or powerful hardware in order to be implemented, they are easier to maintain and update, and they tend to be easy to implement together with external information. On the other hand, they generally suffer from poor generalizability and poor disambiguation ability.

Early on, an important step to make the systems more responsive to changes in the context was to employ semantic categories and rules in semantic-grammar models (Kamath et al., 2018). For instance, the Rapidly Extensible Language (REL) (Dostert et al., 1969) performs semantic analysis first, to then map syntactic arguments to objects from a a datasets.
Another early system is the Lunar Sciences Natural Language Information System (commonly referred to as LUNAR) (Woods, 1973; Woods et al., 1972), which makes use of a parser based on syntactic and semantic information to represent the meaning of queries written in natural language. In particular, the system makes use of rule-driven semantic interpretation and a domain-specific dictionary to interpret questions and answer them according to a background knowledge base.

The Language Access to Distributed Data with Error Recovery (LADDER) system described by Hendrix et al. (1978) and Sacerdoti (1977) is developed in order to retrieve information from the domain of Navy commands. The parser for this system is built on domain-specific labels rather than grammatical ones.
Thus, for instance, rather than labels such as NOUN-PHRASE or VERB-PHRASE, LADDER uses labels such as `ship specification` and `carry-verb phrase` to interpret queries. For instance, in a question such as *Where is the Kennedy?*, *Kennedy* would not be interpreted as a noun phrase, but rather as a `ship name`, thus leading to more domain-specific representations.

The use of pattern recognition to give more freedom to users with their inputs is a step in the right direction compared to other earlier systems such as Clout, which lacks the acceptance of misspellings, and NaturalLink, which employs menus with pre-defined phrases (Johnson, 1984).

Templeton et al. (1983) describe the End-User Friendly Interface to Data management (EUFID), which employs a domain-specific dictionary and rules in order to map input sentences into a semantic graph. In order to account for incomplete, or mistyped inputs, the system takes into account all the possible partial parses of the input. As described by the authors, this system is mainly semantic, and syntactic conditions are introduced only when semantic ambiguities arise.

In an early survey on the commercial applications on NLP techniques (Johnson, 1984), the authors describe the system SAVVY as the pioneer in the field of micro applications for the industry. SAVVY is a semantic parsing system, developed around pattern-matching, with patterns specifically built in order to accept even ill-formed queries.

Despite some of the early successes with this kind of system, they present many issues. Generally speaking, rule-based systems take a lot of manual effort in order to be implemented: every rule has to be created and then checked so that it does not cause any contrast with pre-existing ones (Chiticariu et al., 2013). In addition, these systems usually do not generalize well, since it is hard to rule out every possibility inherently present in natural languages (Waltl et al., 2018).
Furthermore, once a domain-specific dataset or set of patterns is implemented together with the parser, it is hard to re-implement it for a different task (Kamath et al., 2018), which hinders the access to previously generated resources.

2.2.2 Statistical Systems

One step toward improvements in terms of generalization is represented by statistical models. While rule-based systems employ pre-defined rules to map meaning representations, statistical systems learn the most likely representation for utterances using a background dataset as a source of knowledge.

The CHILL system described by Zelle et al. (1996) represents one of the earliest attempts at developing a model based on learning probabilities for semantic parsing. In particular, the training is done on a corpus of pairs `question - query`, which is the most common method implemented to train such systems.
One of the main issues in the CHILL system is that it needs a certain number of (non-trivial) manually annotated natural language utterances in order to be trained effectively. Furthermore, it is based on a hand-built domain-specific lexicon as a source of external knowledge. Both these issues have been looked into, in order to find whether the processes could be automated.

The work by C. A. Thompson et al. (1999), for instance, employs the use of selective sampling techniques to reduce the number of manual annotations needed, by filtering out less informative parses.

In C. Thompson (2003), on the other hand, the authors focus on automating lexicon acquisition by using the training set and semantically labeled trees. Furthermore, the parser learnt using the latter system can be easily ported to languages other than English.

All these works use Prolog (Giannesini et al., 1986) as a formal language to represent meaning. By contrast, the system described in L. S. Zettlemoyer et al. (2005) is trained on pairs of `question - logical forms`, where the logical forms are represented by lambda calculus expressions. In this work, a probabilistic grammar is automatically learned by mainly using domain-specific lexical features. This grammar is then used to assign probabilities to the different possible parses in case ambiguity arises.
In particular, the grammar used is a probabilistic extension to *combinatory categorial grammar* (CCG) (Steedman, 1996, 2000). The authors further improve on their results in another work (L. Zettlemoyer et al., 2007) by making the CCG more flexible in which kind of inputs it accepts.

The systems hereby described are all *fully supervised*, that is they are trained on pairs of natural language sentences and their mappings into some kind of formal language. Other solutions of *weak supervision* have been explored through the years.
For instance, efforts have been made in training statistical systems by pairing natural language utterances with the *result* of their corresponding logical queries on a dataset (rather than with the query itself) (Berant et al., 2013; Clarke et al., 2010; Kwiatkowski et al., 2013; Pasupat et al., 2015), or with the behavior they are supposed to cause in systems (Artzi et al., 2011; Goldwasser et al., 2011).
The main benefit of these models based on denotations is that it is usually easier to manually create the needed datasets rather than annotate natural language utterances to their logical form. For instance, in a QA environment this would mean annotating a question with its answer from a knowledge base, rather than the query it should be mapped to. Furthermore, while there exists many possible logical languages, denotations are usually less specific, which means that these datasets can be reused for other systems as well.

Among these, of particular interest is the model proposed by Kwiatkowski et al. (2013). The parser adopted by the authors is made of a two-step process: first, domain-independent CCG parse trees are built and then these trees used to write under-specified logical forms; then these logical forms are matched to a

domain-specific ontology by replacing under-specified constants with constants representing concepts from the ontology. A model is then trained by using these parsers, together with the final answers to the questions.
The main contribution of this work is the possibility of reusing the results from the CCG parsing step for other tasks and/or other domains.

Other methods of weakly supervised and unsupervised statistical model are further described by Kamath et al. (2018).

2.2.3 Deep Learning Systems

In recent years the use of neural systems has become more and more common for many NLP-related tasks. This applies to semantic parsing as well.

The use of neural networks (NNs) to solve this task has many advantages: most importantly, it avoids the need for explicitly defined lexicons, templates and features, thus making the model easier to get going while also being easily adaptable to other domains. Furthermore, this means that all the model needs is a corpus of examples to be trained on. Finally, it makes full use of vector space models to represent words, which comes with all the advantages of this kind of model.
On the other hand, this also means that these models come with all the issues related to neural models, such as the high amount of data needed in order to train them, and the difficulty to update and maintain them.

There are two main paradigms in deep learning systems for semantic parsing. In the first one, this task is tackled as if it were a translation task: that is, the neural systems are trained on a set of pairs `utterance - logical form` to learn how to map natural language utterances to their corresponding logical form. In the second one, neural modules are implemented to solve specific subtasks for the general aim of semantic parsing.

Semantic Parsing as Translation

A traditional bag-of-word approach with word embeddings would not be beneficial for the task of semantic parsing, since information about the order would be lost and the semantics of an input is strictly related to its surface form. Thus, one possible solution is to make use of recurrent neural networks (RNNs) (Elman, 1990), which allow for a better representation of sequences of words in a vector space by retaining information about the original structure of the input.
One such architecture is Long-Short Term Memory (LSTM), which is useful to

learn long-term dependencies by representing information from the context preceding a given word (Hochreiter et al., 1997). LSTM makes use of *gates* to compute whether a piece of information should be kept or deleted, and to what extent. Similarly, a bidirectional LSTM (BiLSTM) takes into account both the preceding and the future context of a word.

In Dong et al. (2016), for example, authors train a model to map natural language inputs to a logical form representation by employing a Seq2Seq model: an encoder first maps the input to a vector representation; then a decoder maps this representation to a logical form expression. Each of the encoder and decoder have LSTM units, and an attention mechanism is implemented from the decoder to the decoder to make full use of the hidden vectors of the input sequence.

The integration of character-level representation is explored by van Noord et al. (2020), together with a contextual language model, to improve semantic parsing. According to the results found by the authors, these improvements are larger than those resulting from the implementation of individual sources of linguistic information (e.g., POS, dependency parses, or semantic tags) together with word embeddings.

Nevertheless, many issues arise in working with these kinds of systems. First, they are extremely data-hungry. While some languages (especially English) can benefit from a large amount of data freely available on the Web, the same cannot be said about other languages. Similarly, the recollection of training data for specific tasks and specific domains can represent a very cost-intensive effort.
Additionally, certain words or combinations of entities-relation may appear with a low frequency in the dataset, thus affecting their final representation.

Neural Modules for Subtasks

The models proposed by Bao et al. (2016) and Sorokin et al. (2018) aim at mapping natural language questions to semantic queries through a series of staged actions. One of the issues to solve in order for such a model to be effective is the implementation of a system to find the correct final graph among all the possible inferential chains that may arise given an input question.
The use of linguistic information on top of these systems can be extremely useful in solving this issue. Bao et al. (2016), for instance, propose the implementation of a *type constraint*: from the input question, in fact, the model could infer the type of answer, and this information can be used to prune the list of possible re-

lations. For instance, in the question *Who painted the Mona Lisa?* the *who* is already defining that the answer is some kind of person. Furthermore, entity types can be used to further filter out impossible relations by making use of information about domain / range, thus reducing the amount of data needed to train the model.

In the work done by Sorokin et al. (2018), this is further implemented in a Gated Graph NN (GGNN), which is used to compute a final representation for the resulting graph. Most interesting, this final graph representation incorporates information from both a KB and word embeddings.

Once again, these models need a large amount of annotated data to be trained on, something that is not easy to find for specific languages and specific domains. But what this kind of model shows is that neural networks can be implemented to solve smaller, narrower tasks, and the outputs of these task can be combined through other techniques to build the final semantic representation. The main benefit of such a system is that the amount of data needed is lower than that needed to train a neural model for full mapping of semantic representations.
In particular, since the rise in popularity of freely available pre-trained contextual models (e.g., BERT) this can be easy to implement via fine-tuning of a small task-specific dataset which is easy to obtain or annotate (M.-T. Nguyen et al., 2020).

2.3 Neural Word Embeddings

Recently, the use of short, densely populated **embeddings** to represent words has seen an increase in popularity in many NLP tasks. Since the work by Bengio et al. (2003), these embeddings are usually built by implementing a Neural Network, despite some *count-based* implementations still being used, with the most popular one being GlOve[5] (Pennington et al., 2014).

These models have also been improved by the many efforts made regarding neural network architectures, in particular since the implementation of models such as RNN (Melis et al., 2017; Zaremba et al., 2015), which allows for outputs at each epoch to be used as inputs for next epoch. This means that information does not get lost along the model, and that historical information is retained.
Another such model is the convolutional neural network (CNN) (Dauphin et al., 2017), in which a small matrix of weight is passed over the inputs to get only the most salient information from each subsection of the input. While mainly used

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

for computer vision tasks, CNN has been used in NLP as well, in particular in combination with LSTM (Chiu et al., 2016; J. Wang et al., 2016).
Generally, neural word embeddings are learned by training the model on the task of Language Modeling, that is the prediction of the probability of a given word sequence. The final objective is for *acceptable* sequences to have a high probability, and for *unacceptable* sequences to have a very low probability. While this task might seem rather simplistic, it actually lets models learn a lot about language use, as proven by the many applications of LMs (Qiu et al., 2020b).

Indeed, one of the main benefits of these models is that, since they are trained to solve a general task, the embeddings they create can be used for other tasks as well, even on their own (Petroni et al., 2019). Furthermore, they can also be used on more specific tasks by implementing a phase of *fine-tuning*, during which the vectors are re-adjusted on a smaller dataset in order to solve more narrow tasks.

Generally, word embeddings are classified into one of two families: **static embeddings**, in which one fixed vector is learned for each word in the vocabulary; and **contextual embeddings**, where the model learns, for each word, several embeddings reflecting the word's use in different contexts.

2.3.1 Static Word Embeddings

One of the earliest and most influential models for word embeddings is word2vec[6], proposed by Mikolov, Yih, et al. (2013). The main intuition behind word2vec is based on the work on **self-supervision** (Bengio et al., 2003; Collobert et al., 2011), that proved how a neural network could be trained to predict the next word in a sentence using only a corpus as a training dataset. The main benefit of this architecture is that it overcomes the need for annotated training data, and still encodes embeddings representing useful information about tokens.
Word2vec makes use of two separate algorithms: the Continuous Bag of Words Model (CBOW) and the skip-gram with negative sampling (SGNS), that learn embeddings by using two different tasks. In CBOW, given a right and left context, the model tries to predict the central word. In SGNS, on the other hand, the process is the opposite: given a word, it tries to predict the words in the right and left contexts.
The embeddings that are created represent words, both in terms of semantics and in terms of syntax usage. This is proven by the applications of word2vec in sev-

[6]https://github.com/tmikolov/word2vec

eral tasks, such as NER (Das et al., 2017; Horn, 2017; Sienčnik, 2015), sentiment analysis (Acosta et al., 2017; H. Liu, 2017; Petrolito et al., 2018) and syntactic parsing (Ling et al., 2015).

One of the main issues with word2vec is the handling of out of vocabulary (OOV) words, which cannot be encoded in the same way as other words as they never appear in the training corpus. Furthermore, a rich morphology can affect the model, since the different forms for nouns and verbs could occur with different frequencies in a given corpus. A possible solution to both these issues is the one implemented in fasttext[7], as described in Bojanowski et al. (2017) and Joulin et al. (2017).

In fasttext each word is in fact represented by both itself and a bag of constituents of length n. For instance, the word *Gioconda* with $n = 3$ would be represented by <gioconda> and <gi, gio, ioc, oco, con, ond, da>. Then, embeddings are learned using a skipgram model for each of these constituents and the word itself. In case the word has never been seen before, it is represented by a function of its constituents.

Nevertheless, the main issue with static embeddings is that each word (or constituent) is represented by one fixed vector. This does not reflect the natural use of language, where a word such as *bank* has two different meanings based on the context it is used in: *going to the **bank** for a loan* and *the river **bank** is full of flowers*. These two different meanings should be represented by two different embeddings, which is the main reasoning behind the development of *contextual embeddings*.

2.3.2 Contextual Word Embeddings

In contextual embeddings, each token is represented by a function of the input sequence, so that its modeling depends on the sense in which the word is being used (Q. Liu et al., 2020). One of the precursors of modern contextual embedding methods is the work presented by A. M. Dai et al. (2015), in which a sequence autoencoder is implemented to improve sequence learning.

Among more recent works in the field is the Embeddings from Language Models (**ELMo**) (Peters et al., 2018), in which the input raw vectors are first computed based on a character-level CNN, then passed through two layers of bidirectional language model (biLM).

[7] https://fasttext.cc/

A biLM works in two steps: first, a forward pass uses the context before a word to predict the word itself; a backward pass, on the other hand, predicts a word based on the context following it. The representations resulting from the first layer of biLM are fed into the second biLM layer, which outputs another set of representations. The weighted sum of these two intermediate representations and the input raw character-based vectors is the final ELMo representation.

This model greatly helps retaining information about context in a sentence. It has seen application in many tasks such as word sense disambiguation (Kutuzov et al., 2019), NER (Dogan et al., 2019; Sheikhshab et al., 2018; Zhai et al., 2019) and semantic parsing (Einolghozati et al., 2019; Oren et al., 2020; Rongali et al., 2020).

A groundbreaking innovation in NLU is represented by the proposal of the Transformer model, described by Vaswani et al. (2017), which makes full use of the concept of **self-attention**.

The main idea behind self-attention is that, given an input sentence, the representation of each of the tokens should be associated to the representations of other tokens in the same sentence. For instance, given the sentence:

(4) *When I visited the Louvre and saw the* ***Mona Lisa****, I found* ***it*** *smaller than I expected.*

the embedding for *it* should, in some way, encode the fact that it refers to the *Mona Lisa*, rather than to the Louvre. With self-attention, during the processing of each token, the model looks at other positions in the input sequence, then it gives more or less attention (in terms of weights) according to the desired results.

In the Transformer model, self-attention is applied at each step of the training process. The results are incredibly good, with the base model outperforming other models for many tasks, while also being relatively fast to train since it is able to employ parallelization (Vaswani et al., 2017).

This, however, comes with a high price in terms of data and hardware. One of the benefits of the models, though, is that a big model *pretrained* on a general language task (such as word prediction) over a large amount of data, can be fine-tuned on a downstream task, that is going through a smaller steps of re-training on a more specific task to finely adjust the embeddings in order for them to be usable in that specific task. This process is also known as *transfer learning*.

Transfer learning has become a staple in recent NLU researches since the raise in popularity of Transformer models, such as BERT, proposed by Devlin et al. (2019), and the Generative Pre-trained Transformer 3 (GPT-3), proposed by T. Brown et al. (2020), especially with the raise in availability of pre-trained model on the web[8]

BERT is one of the most popular choices in this regard. It employs a Transformer model that looks over the whole sentence both from left and right of each token during training. In order to solve the issue of OOV words, the original BERT employs a WordPiece tokenizer, where each word is split into subword units learned statistically over the vocabulary of the trained corpus (Y. Wu et al., 2016). These subword units may represent morphemes, but this is not always the case (Westhelle et al., 2022).

Furthermore, BERT is trained on two separate tasks at the same time: Masked Language Model (MLM), which first replace around 15% of the tokens with a `[MASK]` token, then tries to predict those replaced units; and Next Sentence Prediction (NSP), that predicts whether a sentence might follow the previous one.

Over the years, many models have been proposed based on the BERT architecture, for instance:

- RoBERTA (Y. Liu et al., 2019), which removes the next-sentence prediction while implementing dynamic masking, where the masked tokens change during training;
- XLNet (Yang et al., 2020), which introduces permutation language modeling, where all tokens are predicted in random order;
- SciBERT (Beltagy et al., 2019), based on base BERT and pre-trained on scientific publications.

These models have been implemented for various tasks, generally achieving state-of-the-art results. Some examples include: NMT (Clinchant et al., 2019; Zhu et al., 2020); text classification (Chang et al., 2019; Garg et al., 2020); semantic parsing (He et al., 2020).

As previously mentioned, despite the great results in many applications, issues are still present in terms of bias and lack of grounding in any sort of knowledge of the world (Bender et al., 2021). Similarly, while multilingual support is present for several BERT models, the results are not coherent across all languages (S. Wu et al., 2020), and the languages supported still do not cover for many of the low-resource languages around the world (Joshi et al., 2020).

[8] For instance, the platform Hugging Face Transformers provides APIs to quickly access and train models, https://github.com/huggingface/transformers.

2.3.3 Integration of Structured Knowledge into Embeddings

Recently, there have been many efforts regarding the injection of knowledge from an external, structured source into word embeddings, in order to solve some of the issues inherently present in distributional models (Roy et al., 2021). In Sheth et al. (2017), the authors describe three main situations where the use of structured Knowledge Bases can be used to improve neural models:

- in cases where hand-labeled data is not readily available;
- in case the text to be recognized present complexities, for instance in the type of entities to classify, or in the inherent subjectivity of some features (e.g., emotions and intention);
- in cases where multimodal data is present.

The need to bridge the gap between the two paradigms of AI is generally referred to as *neural-symbolic integration* (Bader et al., 2005), and in the field of distributional representation the attention to this topic has been rekindled in recent years by the need to enhance language models so that they can support actual, explicit grounding to real-world concepts (Peters et al., 2019). The main benefit of doing this is that the contexts of use, inherently encoded by word embeddings, can benefit from explicit *semantics* and *syntax* knowledge, extracted using an external source of data.

The analysis presented by H.-Y. Wang et al. (2017) proves that the inclusion of relational information in distributional models improves the final results for tasks such as word similarity and analogical reasoning, thus making the process of integration an improvement both in terms of final results, and in terms of *climbing the right hill* (Bender et al., 2020).

While many approaches exist, Emerson (2020) introduces three paradigms for grounding structured information into distributional models:

1. by training a base distributional model and then combining it with a grounded model;
2. by linking distributional and grounded features through correlations;
3. by jointly learning distributional and grounded parameters.

The author admits that the last option would be the best option to ground corpus data, but it is also the most computationally intensive one to recreate (Emerson, 2020).

Another classification of the current paradigms for the infusion of knowledge in Deep Learning for NLP is the one proposed by Sheth et al. (2019). The authors investigate three paradigms for knowledge infusion: *shallow*, *semi-deep* and *deep*.

In shallow infusion, the structure of knowledge is disregarded, as it is first transformed into a flattened intermediate form, so that it can be directly injected into the learning model without any significant change. The external information is injected by first training a model on a large corpus, and then used as an input for a second, task-specific, model.
In the semi-deep paradigm, alternatively, the addition of structural (e.g., dependency relations) or symbolic (e.g. attention probability) information is used to resolve some mismatches that might be present in a deep net. This is effective in learning features that the model is unable to learn from text alone.
Deep infusion paradigm, finally, is based on the combination of representations learned from a deep neural network with semantic information from KGs, that can be used to reveal patterns that might have been missed. In particular, the author suggests that this kind of infusion should be applied within the latent layers of a neural network to boost learning (Emerson, 2020).

In this work, the paradigms are divided in two main categories, according to how structured knowledge is integrated in distributional representations: it can be done with regards to a specific task, for instance relation extraction; or external knowledge can be injected directly into the training steps of a LM, which would make the final results usable in many different environments and settings, and more robust to changes in language/domain of use.

Task-Specific Integration

Over the years, the integration of external sources of knowledge together with word embeddings has been explored for several task-specific applications. These paradigms for integration are generally implemented at the level of the models, or at the level of the training dataset.

Model-level integration Passos et al. (2014) propose the use of a lexicon together with a space vector model to improve the results on NER. In particular, on top of a the word prediction task of the Skip-gram model, this system features a classification task over possible lexicon classes, derived from Wikipedia pages.

Celikyilmaz et al. (2015) describe two models specifically designed to enrich word embeddings from word2vec with structured information for the semantic parsing to natural language queries. The first is Context and Entity Constrained Model (CECM), which puts implicit constraints on the context of the word by querying a dataset. More precisely, this model maximizes the log likelihood of

each query token given both its context words and any other entities present in the question itself, according to an external source of knowledge, for instance Freebase.
The second model is a Relation Encoding Model (REM), which further encodes relational data into the previous CECM method inspired by the TransE model (Bordes et al., 2013). In particular, the assumption is that given a triple `(s, r, o)` present in the Knowledge Base, the vector representing the relation realization `r` should correspond to a translation of embeddings from subject realization `s` to object realization `o`. Thus, by enforcing the equation $s+r \approx o$ the embedding representing `r` would effectively retain information about the relation it realizes.
To implement such a model, a precise Entity Linking and Relation Extraction process is needed, which is not always possible to include for every language and every domain.
While the TransE model is used to create Knowledge Base embeddings, CECM and REM are used to build word embeddings. One of the issues related to this technique is that for it to be implemented it has to be assumed that all the entities present in a knowledge base have enough examples in linguistic data for them to be expressed in the form of meaningful embeddings.

Another method to integrate linguistic information together with structured knowledge is presented by Long et al. (2016), who propose the use of entities' textual description, which might be available in certain datasets, to better initialize the TransE model. While this solution might work with some specific knowledge bases (e.g., WordNet, Freebase and Wikidata), the assumption that entities' textual description are available of all datasets is not always true.

Data-level integration While the solutions previously described are integrated during the training step of neural models, another set of solutions operate at the corpus level, by creating better and more robust examples in order to improve pre-training and fine-tuning.

As mentioned already, one common issue in these neural systems is how they handle rare words, or rare features. While one possible solution to rare words is to use subword tokenization rather than word-level tokenization, this does not help with having more robust examples for other features, for instance entity-relation pairs. Because of this, efforts have been made in the use of knowledge injection to have better examples for such rare phenomena.

Jia et al. (2016) explore the application of a *recombination* process to create more robust training examples. In particular, given a question as an input, any

entity is first substituted by its *type*. Then, each type is replaced by each of the possible entities once, creating a new set of examples.
This process, visualized in Example 5, can leverage either an external lexicon or contextual information for entity recognition.

Original Examples

who painted [the mona lisa]?

when was [guernica] painted?

↓

Entity Replacement

who painted `painting`?

whan was `painting` painted?

↓

Recombination

who painted the mona lisa?

when was the mona lisa painted?

who painted guernica?

when was guernica painted?

(5)

This would, in turn, improve how much the system can generalize by actually providing a set of examples reflecting the typology of entities. Indeed, in order to complete such a process, the corpus needs to be augmented by some sort of entity classifier and/or entity linker.

Dimitrakis et al. (2019) describe how resources, such as lexicons, can be exploited to reformulate questions in Information Retrieval tasks. Techniques for reformulation based on the redundancy approach for QA have also been explored by J. Lin (2007).

Similarly, an external source of structured knowledge can also be used to automatically label data. This, in particular, has been proven particularly useful for the task of *relation recognition*. In fact, while information about relations may be learned by a neural system by training a NN on a classification task, the annotated data needed for such a training effort is not always easy to find, in particular for specific languages. Furthermore, the manual annotation of such a dataset requires a huge effort by experts in specific domains.

One possible solution to this issue is to apply *distant supervision*, as first proposed by Mintz et al. (2009). The main idea behind this technique is that, given a set of triples `(subject, relation, object)` from a KB, and a set of possible surface forms that the subject and the object can take in NL utterances (*mentions*), it is possible to automatically label any sentence containing a mention of the subject and a mention for the object as being representative of that specific relation. The main assumption is thus that, if a sentence shows both a mention for the subject and a mention for the object, it should be given for granted that the sentence would also realize in some way or another the relation between the two.
Thus, given the triple `(mona-lisa, created-by, leonardo-da-vinci)`, a list of possible mentions for `mona-lisa` such as `[mona lisa, gioconda]`, and a list of possible mentions for `leonardo` such as `[leonardo, leonardo da vinci]`, the following sentences would be labeled as representing the relation `created-by`:

(6) Leonardo painted the Mona Lisa
The Mona Lisa, one of the most famous paintings by Leonardo da Vinci, is currently displayed in The Louvre.
Leonardo da Vinci is especially renowned as the author of the Mona Lisa.

Distant supervision is easy to apply to any combination of DB/corpus, given that a mapping from entities to mentions is available, as proven by Mintz et al. (2009), who uses Freebase as a background DB, with a named entity tagger to link mentions to entities.
The distant supervision paradigm has since been used for other applications. For instance, W. Xu et al. (2013) apply this technique to Knowledge Base completion.

Baldini Soares et al. (2019) propose to extend distant supervision by applying distributional hypothesis to fine-tune a Transformer model.
Given a dataset automatically labeled by levering information from an external dataset, the authors define a relation statement as a triple $(\mathtt{x},\ s_1, s_2)$, with `x` being a sequence of tokens, and s_1 and s_2 being indices delimiting entity mentions in `x`.
As opposed to the model described by Mintz et al. (2009), this system does not train a classifier, but rather it learns to map a relation statement to a fixed-length vector representing the relation between the two entities of interest. This is done by maximizing the probability that two relation realizations between the same two entities represent the same relation in the Knowledge Base.
Finally, the sentences representative of relations are modified, so that entity mentions *may* be replaced by a special `[BLANK]` symbol. This is done so that the

system does not simply relearn the entity linking system, which would otherwise perfectly solve the task.
By employing this method, relations can effectively be encoded in the same space as word embeddings, by making full use of the data available on the Web while still leveraging information from a structured source of data. In particular, the system is applied as a fine-tuning step to BERT base model.

One further improvement to the model defined by Baldini Soares et al. (2019) is the application of pronoun resolution in order to cover for sentences where entity mentions are implied. This is extremely common in Italian data, and would improve the system greatly.
One of the main issues with distant learning is that its basic assumption that every sentence mentioning a subject and an object of a relation also shows realization of a relation is a strong one. Furthermore, this process tends to create extremely noisy results if the output is not checked, since any error in entity linking will also propagate in the final annotated output.

Another possibility is to leverage Natural Language generation to build task-specific datasets from KB triples. The main idea is to convert triples into sentences by using relation templates, and has been applied in many different environments, such as story generation (Guan et al., 2020), QA (Ma et al., 2020; Ye et al., 2020) and relation classification (Bouraoui et al., 2019). While the basic approach to this kind of system is to handcraft templates for sentence generation, distant supervision might also be implemented to automatically mine templates (Bouraoui et al., 2019; Safavi et al., 2021; Ye et al., 2020).

The use of linguistically motivated templates can help improve distant supervision by removing noisy examples on the basis of structured, grounded information.

One general issue of the presented methods is that, while they are useful in solving specific issues, the embeddings are not re-usable for more general tasks. A better solution in this regard is to inject knowledge directly into the Language Modeling task, so that it can then be leveraged together with linguistic information, and reused for different applications.

Knowledge-enhanced Language Models

In recent years, many efforts have been made in trying to leverage structured source of knowledge in the pre-training step of LMs. While this can generally be done by connecting word embeddings to information from a KB by employing an **entity-aware model**, new Transformers models have given new possibilities in this regard, in particular through the usage of the attention module. As such, **knowledge-enhanced transformer models** can be used to improve the representation of meaning by integration with an external source of knowledge.

Entity-aware Language Models Bian et al. (2014) integrate morphological, syntactic, and semantic knowledge into word embeddings through several steps. On one hand, morphological information are integrated by exploiting either a root/affix or syllable dictionary, in place of the standard word dictionary.
This helps both introduce external knowledge into the system, while also reducing the overall vocabulary size and helping with unseen words. This is not too dissimilar to the word-piece tokenization employed in BERT, but while original BERT learns its WordPiece tokenizer statistically, this method can be guided by linguistic knowledge to split words into actual morphemes.
Syntactic and semantic information, on the other hand, are implemented by *extending* the original word vector to include further information. For instance, an *entity vector* would be a one-hot vector containing non-zero values only in positions which would indicate that the word belongs to certain entity categories. Relationships and POS value could be represented in a similar way as well.
While this method proposes an interesting and simple way to integrate external structured source of knowledge, the extension of the original word vector would affect the hardware usage, making this process less cost-effective than other proposals where embedding dimensions do not change. Furthermore, the information would get implemented in different parts of vectors, without any kind of communication between them.

Another integration of external knowledge for word2vec is the Relational Constrained Model (RCM) presented by M. Yu et al. (2014). RCM refers to an objective in which words are predicted based on some externally represented relationship (e.g., a property from a knowledge base). This model learns embeddings by jointly training the `cbow` objective from word2vec, together with RCM, thus integrating external knowledge into the basic word2vec contextual representation.

The method presented by Faruqui et al. (2015) first learns embeddings based on distributional information, then post-process them based on semantic lexicons, so that words linked by any relation end up having similar word representations.
In practice, once embeddings are learned, they are retrofitted so that they are both still close to the original learned vectors, while also being close to adjacent vertices given a semantic graph representing a background knowledge base. While

the authors experiment with three lexicons as possible sources of external knowledge (a paraphrase database, Wordnet and Framenet) the model could be theoretically implemented with any kind of knowledge base with the help of an entity linking module.

Ahn et al. (2017) propose to combine symbolic knowledge from a knowledge graph with a RNN LM. In particular, this model is based on connecting words to a fact-based topic knowledge, where each fact is represented by a triple in which the word realizes either the subject, the object, or the relation. One of the main issues of this proposed model is that it is based on a simple process of string matching, which might not be enough for entity linking (especially in case of complex terminology compounds found in specific domains).

The model proposed by Yang et al. (2017) explores the use of references in the training of Language Models. In particular, the authors propose three kinds of references: references to a list (for instance a list of ingredients given a recipe); references to a database; and references to other elements in the document.

As mentioned by Peters et al. (2019), one of the main issues of the entity-aware language models is that they require a full annotation before training step. For instance, the method proposed by Yang et al. (2017) requires annotation for either one of the reference models, which might not be accessible given certain languages. This necessity for high-level annotation can be reduced by employing more weakly supervised models.

Knowledge Enhanced Transformers Many efforts have also been made in injecting information from Knowledge Base into transformer models.

Peters et al. (2019), for instance, present KnowBert, in which the attention mechanism is applied to embed knowledge bases into LMs. One of the main contributions of this method is that the entity linking module that is used to connect entity mentions to the entities they represent in the KB is learned jointly with the language modeling task.
Then, once the entity linking process is completed, the information extracted from the entity in the Knowledge Base is injected into the mention-span representations built by BERT.
Finally, word representations are recontextualized according to the information from the entity representation by making full use of a Multi-Head Attention as described by Vaswani et al. (2017), between the word-piece representations and knowledge-enhanced entity span vectors.

This information is then mapped to one-hot vectors, which can then be inserted into the self-attention layer of BERT. Furthermore, knowledge is shared among the sub-networks through a layer of topical attention based on a trainable task-specific query vector.

J. Zhou et al. (2020) propose Linguistics Informed Multi-Task BERT (LIMIT-BERT) as an attempt to inject linguistic knowledge into LMs. In particular, the core of this model is Multi Task Learning (MTL) (Caruana, 1993) to make representations transferable between language-specific tasks.
In particular, every sentence in the training dataset is first automatically labeled with POS tags, syntactic information and semantic information.
Then, during the language model training process, this information is fully exploited in several task-specific layers, each scoring its own objective. The loss for each layer is taken into the account for the overall loss of LIMIT-BERT.

K-Adapter is an attempt to integrate both factual and linguistic information, described by R. Wang et al. (2021), and it further shows a lot of flexibility by integrating independent learning in its modules.
The main idea behind K-Adapter is that of using *adapters*, that is knowledge-specific, independently trained modules with few parameters in which the outputs of the intermediate hidden states of the pre-trained model can be inserted. In particular, the authors propose a *relation classification* adapter for factual information, and a *dependency relation prediction* adapter for linguistic knowledge. The model is tested on RoBERTa, but it can be theoretically integrated with any transformer model.

Bai et al. (2021), on the other hand, propose Syntax-BERT, a model specifically built in order to introduce structured information about syntax in BERT embeddings. In particular, the authors propose to exploit the information from a dependency graph in the form of masks which are then attended to through an attention layer.
Despite all these models are extremely efficient in integrating external knowledge into transformer models, while also showing extremely good results, they all ask for a huge amount of data and high costs in terms of hardware in order to be implemented, and maintained. Furthermore, since their cores lie still in the LM part of the system, they are still largely non symbolic.

2.4 Graph-based representations

As mentioned already in Section 1.4, the use of graph-based structures to represent information is particularly useful in those cases where the interconnectivity of elements is as important as the elements themselves. Furthermore, in the field of NLP the application of graphs has been proven to be extremely beneficial for several reasons (Nastase et al., 2015):

- graphs can be useful in revealing regularities and patterns in data;
- graphs can be inspected by human's eyes, providing insights that can be applied to automatic methods.

The authors also underline some of the issues in the implementation of graphs for NLP tasks, more specifically

- many graph-based algorithms do not scale well to large data sizes;
- data in NLP should be represented by streaming graphs (i.e., graphs whose data changes over time), which represent a difficult implementation of data

Despite the issues still present, graphs represent an important topic in natural language representation, as it is shown by the presence of several books and surveys written on the topic (Hamza Osman et al., 2020; Mihalcea et al., 2011; Mills et al., 2014; Nastase et al., 2015).

2.4.1 Graph-Based Text Representations

As mentioned in Section 1.4, the definition of a graph-based model for knowledge representation offers much flexibility in the structure used for the representation. This applies to natural language representation as well, where nodes and edges can be made to represent language-related information of various forms, according to the needs of a specific task (Nastase et al., 2015).

For instance, nodes might be used to represent units at the same hierarchical level: this is the case, for instance, of dependency and constituency graphs, where every node generally encodes a singular token. But we might also have elements of multiple hierarchical levels, for instance nodes might represent documents, sentence, and words, in a tree-like structure.

Generally speaking, node can encode text units of various sizes such as words, collocations, word senses, sentences or documents. Edges, on the other hand, could encode relations such as co-occurrence, collocation, syntactic structure, lexical similarity or domain-specific relations derived from ontologies.

For instance, in one of the first works to apply graph-based methods (Salton et al., 1997), authors represent documents as a series of paragraphs. Both documents and paragraphs are connected by edges, whose weights represent the similarity scores between the texts, computed on the basis of vocabulary overlap.

L. Wu et al. (2021) introduce two major approaches to graph construction in the field of NLP: static graph construction and dynamic graph construction. While the former deals with the construction of the graph structures during preprocessing (generally by leveraging existing parsing tools or predefined rules), the latter aims at dynamically learning the graph structure on the fly, without resorting to human or domain expertise.
In the following sections we introduce some of the graphs that have been implemented in the literature for NLP-specific tasks.

Co-occurrence Graphs In a co-occurrence graph, the structure is built to capture the co-occurrence relation between words, that is the frequency of two words co-occurring within a fixex-size context window. The idea of representing a language unit in terms of co-occurrence context is also at the basis of the distributional hypothesis (see Section 2.1). The starting point of such a graph is generally a co-occurrence matrix M of a given corpus C, where $M_{(i,j)}$ is equal to the number of times the tokens i and j appear together in C given a context window n. Once this matrix is obtained, L. Wu et al. (2021) underline two main ways of calculating the weights of the edges between tokens:

1. pure co-occurrence frequency, as shown in Christopoulou et al. (2019), Edouard et al. (2017), M. Zhang et al. (2020), Y. Zhang et al. (2020), and H. Zhou et al. (2018),
2. point-wise-mutual information (PMI), as shown in J. Hu et al. (2019) and J. Hu et al. (2020), Yao et al. (2019).

Since this kind of graph is based on the same general principle of distributional hypothesis, they are rarely implemented together with a word embedding model, but they are generally seen as a substitute for the purely distributional approach.

Dependency Graphs A dependency graph encodes syntactical information about a piece of natural language text in the form of dependencies between words. Each node encodes a single token, and the nodes are connected by two different kinds of relations: dependency relations and sequential relations. Dependencies represent syntactical connections between two words (e.g., "subject of", "case of",

Might does not make right!
Right makes right!

	might	does	not	make	right
might		1	1	1	0
does	1		1	1	0
not	0	1		1	1
make	0	1	1		3
right	0	1	1	3	

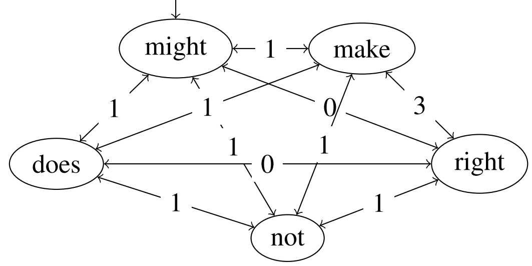


Figure 3: Example of a co-occurrence graph, with context window set at 4.

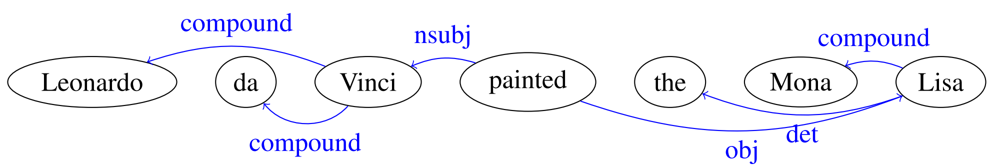


Figure 4: Example of a dependency graph for the sentence *Leonardo da Vinci painted the Mona Lisa*. The sentence was parsed using the CoreNLP online demo.

etc.), while sequential relations connect words adjacent to each other in the given piece of text (L. Wu et al., 2021).
While extracting the sequentiality of tokens is trivial, extracting dependencies is usually done through the integration of off-the-shelf NLP parsing tools (such as Stanford CoreNLP[9], spaCy[10] and TINT[11]. While these tools generally work well and are implemented without many changes, it is important to notice that their results can sometimes be noisy, and their mistakes, if not fixed, can have effects on the whole process. The kinds of error that can be encountered fall in categories such as: assigning the wrong POS tag, inconsistencies in the handling of language-specific phenomena, and the handling of unknown words (Coden et al., 2005b; Mikheev, 1997a).
Dependency graphs have proven successful as a means to integrate structured linguistic information together with distributional representation (Bastings et al., 2017; Marcheggiani et al., 2017; Vashishth et al., 2019).

Knowledge Graphs While the previously presented graphs include information that is extracted by the texts on their own, for specific tasks it might be useful to enrich the original data with external knowledge bases. As mentioned already (Section 1.4), in recent years graph-based structures for knowledge representation have become the norm on the Web, both for general-domain and domain-specific knowledge. The integration of such knowledge is generally reliant on the task at hand and the dataset one wants to implement, with a particular focus being put on the representation schema that is put in place. In particular, the choice of which ontology to use to represent data in this format is extremely important, and has an impact on the task as a whole.
One of such representations, making use of information from WikiData, is shown

[9]https://stanfordnlp.github.io/CoreNLP/
[10]https://spacy.io/
[11]https://dh.fbk.eu/research/tint/

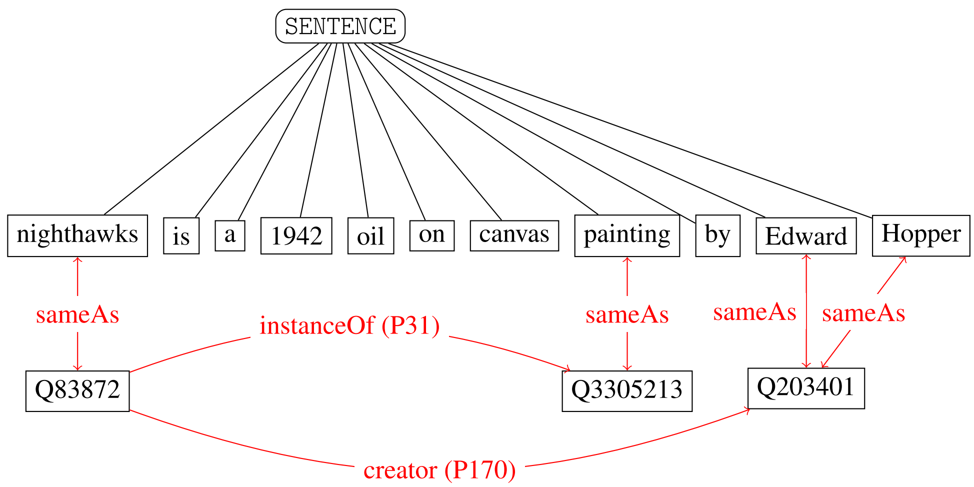


Figure 5: Example of a knowledge graph representation for the sentence *Nighthawks is a 1942 oil on canvas painting by Edward Hopper* and the entities it refers to according to Wikidata.

in Figure 5.

The usefulness of such integration lies in the fact that, in doing so, information from both the text and a grounded source of knowledge can be encoded in the same space so that the model can leverage information from both sources, which makes this kind of graph representation of particular interest for the topic of neuro-symbolic learning. This is particularly useful for tasks dealing with facts and reasoning, such as Question Answering, but the process relies on complex subtasks, such as entity recognition, relations extraction and word sense disambiguation, which add additional complexity to the final system.

2.4.2 Applications

Since the early applications of graph-based representation in semantic networks (R. Quillian, 1968), graphs have been widely implemented to represent texts, in particular as a way to solve the issues inherently present in distributional and Bag of Words approaches. This section defines some of these approaches, by defining the tasks they are implemented for.

Summarization Traditionally speaking, one of the earliest areas of applications for graphs in NLP has been text summarization. The models for text summarization generally employ graphs where nodes represent text passages, rather than tokens, and edges representing similarity between said passages.

Salton et al. (1997) show one of such applications: the authors propose graph-based document representation on the basis of inter-document links between passages composing these texts. These passages can be seen as sentences or paragraphs, and they are represented by a vector composed of the weight of the words which are part of the passage. The weights are calculated on the basis of the importance of the word for that specific term for that specific passage. Given these passage vectors, the authors propose to connect passages on the basis of their similarity, with more similar paragraphs or sentences being connected by a stronger (i.e., with a higher weight) edge. The resulting graph can thus be used as a representation of the internal structure of a document, from which the most important (i.e., informative) passages can be extracted by traversing the paths composing it.

In Zha (2002) the authors propose a method for jointly doing keyphrase extraction and text summarization. The graph representation is an undirected weighted graph, where nodes represent sentences. Two sentences are connected in case they share terms, with the weight reflecting the similarity between these two sentences. In addition to this representation, the authors also incorporate a value reflecting sentence vicinity, so that close sentences would get a stronger link. On the basis of this representation, an algorithm is used to aggregate sentences into topical groups, and then ranked according to their saliency score, and selected for the final summary.

More recently, Mallick et al. (2018) first propose the introduction of a tf-idf-inspired weight for cosine similarity between vectors-sentences, and then apply TextRank(Mihalcea et al., 2004) to rank the sentences in order of importance, to then generate a summary composed of the first *n* sentences.

Sentiment Analysis The use of graph-based representation for the task of sentiment analysis has been proposed in works such as Mukherjee et al. (2012), where the authors first extract domain-specific tokens from sentences, and then extract dependency graphs based on these tokens by taking into account immediate dependencies. Then, these immediate dependencies are connected together in case they share connection through some tokens, so that the list of tokens related to domain-specific words can be extracted and expanded. Once these graph-based representations are defined, they can be used as input to a sentiment analysis

model, either making use of a rule-based classification based on a sentiment lexicon, or learning a supervised classifier such as the SVM.
A more refined model is proposed by K. Sun, Zhang, et al. (2019) to solve the task of aspect-based sentiment analysis (ABSA), that takes as input a sentence representing an opinion over an object, and its aspect (i.e., the element in the sentence that is being reviewed). The tokens in the sentence are first encoded as word embeddings learned through Bi-LSTM to allow for contextual information, then, the sentences are represented as dependency graphs where each node defines a token, initialized as its vector representation. The authors then apply GCN over this dependency graph to include structural and explicit linguistic information in the representations of tokens.
B. Liang et al. (2022), in a recent paper, propose an improvement of the model proposed in K. Sun, Zhang, et al. (2019) by integrating external information in the GCN-based model. In particular, the connection between two word-nodes connected in the dependency graph is stronger in case either of the two words is a word expressing sentiment according to the SenticNet knowledge base[12]. Furthermore, the score for specific opinion words is also included in the graph: the weight of the connection between two words also takes into account the sum of the scores of the words connected, so that negative words would have a negative impact on the connection, and positive words would have a positive impact on the connection.

Machine Translation Graph-based representations for natural language utterances have been recently explored for the task of Neural Machine Translation. H. Chen et al. (2017), for instance, propose the representation of a specific sentence as its dependency tree, which is then fed into a tree encoder (Eriguchi et al., 2016; Tai et al., 2015) that captures both information for the word embeddings computed on the text itself, and structural information from the dependency tree. The trees are further used to control the attention mechanism as a source of prior knowledge.
In Bastings et al. (2017), the authors take the Syntactic GCN model (Marcheggiani et al., 2017) to represent sentences, with a BiRNN model to encode the initial token embeddings, for the task of NMT. Marcheggiani et al. (2018) propose an improvement on this model through the integration of semantic information into a GCN model, in particular by integrating the outputs of a Semantic Role

[12] https://sentic.net/

Labeling task, in the form of the arguments of specific verbs. The inputs are thus knowledge graphs with vectors representing tokens and relations representing dependency connections and semantic connections between verbs and arguments.
Similarly, Song et al. (2019) propose the integration of semantic structures for the task of NMT, but instead of the outputs of an SRL module, the work leverages Abstract Meaning Representations (AMR) (Banarescu et al., 2013), and graph recurrent network (GRN) (Song et al., 2018) rather than GCN.
Recently, K. Chen et al. (2022) have proposed the construction of a document-level graph representation to improve NMT. More specifically, the graph is composed of two kinds of nodes, i.e., sentence nodes and word nodes; and four kind of edges, i.e., inter-sentence edge (connecting all sentence within a specific document), sentence-word edge (connecting one sentence and all its content words), intra-sentence edge (connecting subsequent words within the same sentence) and co-occurrence edge (connecting word appearing in different sentences). Feature representations are learned for each node using Graph attention networks (Guo et al., 2019).

Keyphrases Extraction Keyphrases extraction represent one of the main areas of applications of graph-based representation for NLP.
One of the most influential applications in this field is TextRank, the algorithm proposed by Mihalcea et al. (2004) for the task of Keyword Extraction. The authors propose the use of a co-occurrence matrix (Mihalcea, 2004), in which vertices represent words, and each two words are connected by an edge if the lexical units they correspond to co-occur within a window of size N. The authors experiment with various filters, such as limiting which lexical units can be connected by an edge. Once such a graph is built, the importance of each vertex is computed as a score, by taking into account how many other vertices are linked to it, and how important are these connected vertices. This iterative process is initialized with arbitrary values, and then it iterates until it converges below a certain threshold. The use of such scores was initially proposed in the field of Web page analysis, as seen in Brin et al. (1998).

Over the years a number of extensions have been proposed on top of TextRank (Firoozeh et al., 2020). For instance, ExtendRank (Wan et al., 2008) includes weights as a way to better represent information about word co-occurrence. More specifically, the authors extract a graph such as the one proposed in Mihalcea et al. (2004), but where the edges also take into account the similarity of documents in which the words co-occur, so that words co-occurring in more similar documents

will have stronger connections.
Similarly, Florescu et al. (2017) propose the inclusion of a bias in order to give higher probability to words appearing early in a document, and with high frequency. Every word is thus weighted with its inverse position in the document, and if it appears multiple times, it is weighted against the sum of all the inverse positions.

TextRank has also been extended through the inclusion of topical information. In Z. Liu et al. (2010), for instance, the authors propose Topical PageRank: they first compute topic distribution for each word by using Latent Dirichlet Allocation (LDA)(Blei et al., 2003), and the values for these distributions are then included in the TextRank graph so that words that are more relevant to the document's topic will have higher scores. This model takes into account each topic retrieved when computing the scores, which might be computational-intensive. An optimization of said process is proposed in Sterckx et al. (2015), so that a single PageRank is computed for each text, regardless of the amount of topics extracted.
A similar approach is proposed by Bougouin et al. (2013), with the main difference being that while Topical PageRank uses LDA, and thus needs background training data to compute word distributions, this approach groups together similar noun phrases (in terms of overlapping words) to represent a singular topic. Then, the document is represented as a fully-connected graph where vertices are topics (rather than words) and edges represent the semantic strength between these topics. The TextRank algorithm is then implemented to rank each topic.

An extension of the model described in Bougouin et al. (2013) is the one proposed by Boudin (2018). The graph proposed in this work, rather than representing each separate topic as vertices, makes use of a multipartite graph representation, in which each topic is partitioned in the keyphrase candidates it contains. Each candidate is then represented as its own vertex, and is connected to other candidates only if they belong to different topics. The weight of the edges between candidates is computed as the inverse distances between the occurrences of said candidates in the document.

Semantic Relation Extraction The use of graph-based representation for the task of Semantic Relation Extraction has generally been implemented as a means to exploit external linguistic information, in particular since K. Xu et al. (2015) have proven the usefulness of implementing shortest syntactic dependency paths between arguments for relation prediction, and J. Li et al. (2015) observed that representations based on dependency graphs fare extremely well on the semantic relation extraction task. Benefits of dependency trees for relation extraction have also been described by Y. Zhang et al. (2018).

An early implementation of such ideas is found in Culotta et al. (2004). Given a sentence, and the entities interested in a specific relation realized in the sentence, the smallest common subtree containing both is extracted. Each node of the tree is then augmented with specific features (e.g., POS tag, its Wordnet hypernyms, the entity type, and so on). Based on these tree representations, the authors define a kernel function to compute features for the representations given their similarity scores. This approach showed improvements compared to a basic bag-of-words model.

More complex models have since been proposed. For instance, Peng et al. (2017) implement a generalization of LSTMs that can be applied to graphs. The core of this model is a document graph that encodes dependencies between the words in the input text. In particular, this document graph captures both intra- and inter-sentence dependencies on various level (syntactic dependencies, adjacent words and discourse relations). Then, this graph is split into two sub-graphs, one containing left-to-right chains and the other containing right-to-left chains, which are used as inputs to the LSTMs.

The use of such graph representations for relation extraction has also been explored with different models and frameworks. In particular, Graph Convolutional Networks have been proven extremely effective, as seen in Q. Dai et al. (2019), Guo et al. (2019), Marcheggiani et al. (2017), Sahu et al. (2019), and C. Sun et al. (2019).

2.4.3 Vector Models for Knowledge Graphs

As mentioned already, one of the main benefits of using graph structures is the possibility to encode information about *concepts* and *relations* between them in a structured and symbolic way. While this is the main benefit of such structures, it is also one of the main drawbacks: the inherent structural complexity of graphs, in fact, hinders the recollection of information underlying the structure itself, such as inferences (i.e. the extraction of new facts not inherently expressed by the data at hand) (S. Zhang et al., 2019). One of such inferences is, for instance, the prediction and computation of *node similarity*.

In recent years, a potential solution to this issue has been represented by the use of low-dimensional vector space to represent graphs. As it is also the case for

the representation of natural language using distributional techniques (see Section 2.1), this is generally done through *embedding techniques*, that is by representing information about nodes as a continuous list of numbers so that mathematical techniques can then be used to process it, without losing structural information (Bordes et al., 2013) as well as properties in the LPG paradigm.
In recent years, the encoding of KG using embeddings has become a central topic in the integration of linguistic and structured information for NLU. As described in Q. Wang et al. (2017), the typical KG embedding model consists of three steps:

1. entities and relations' representation
2. scoring function definition
3. representation learning

The first step regards the means through which nodes and edges should be represented: while, as mentioned before, nodes are generally represented by continuous vectors (Bordes et al., 2013; Y. Lin et al., 2015; Z. Wang et al., 2014), there have been different proposals for edges representation, such as vectors (Bordes et al., 2013; Z. Wang et al., 2014), matrices (Y. Lin et al., 2015) and tensors (Socher et al., 2013), among others (Q. Wang et al., 2017).
In the second step, a scoring function $f_r(h,t)$ is defined, such that it can be used to calculate the plausibility of the triple (h,r,t). The main idea here is that facts that are actually present in the KG should score higher than those not appearing in it.
Finally, in the third step the representations for entities and relations are learnt, in the form of embeddings, through an optimization process that attempts to maximize the plausibility of seen facts, while minimizing the plausibility of incorrect facts.
Following the current literature (Huang et al., 2021; M. Wang et al., 2021; Q. Wang et al., 2017), in this work the possible methods for Knowledge Graph Embeddings (KGE) are classified into the following categories:

- Translation-based models
- Semantic-matching-based models
- Neural-networks-based models
  - Random-walk based models
  - Graph Convolutional Networks

By following this categorization, the following sections describe some of the efforts made in the field of KG embeddings.

Translation-based models

These models are based on scoring functions that compute the plausibility of facts as the distance between two entities. The main idea is that a relation is seen as a *translation* from the first entity to the second entity. One of the most representative translation-based models is TransE, as defined by Bordes et al. (2013).
In TransE both entities and relations are represented in the same space as vectors. Given a fact (h,r,t), the relation is assumed to connect the head entity and the tail entity such that $h+r \approx t$. The core idea behind this representation has its roots in the linguistic intuition defined by Mikolov, Yih, et al. (2013), previously mentioned in the context of word2vec. The scoring function in this case defined as the negative distance between $h+r$ and t, that is $f_r(h,t) = -|h+r-t|_{1/2}$. The score would be higher for seen facts.
Over the years, the basic TransE model would get to be improved in various way, in particular to solve the inherent issues related to 1-to-N, N-to-1 and N-to-N relations (Y. Lin et al., 2015; Z. Wang et al., 2014): for instance, entities such as *Mona Lisa*, *The last Supper* and *Lady with an Ermine* could all be learned as similar vectors since they are all connected to the entity *Leonardo da Vinci* by the same relation *created by*, despite being different paintings.
One of the models proposed to solve such issues is TransH, as defined in Z. Wang et al. (2014). The main idea behind this model allows entities to have distinct representations when involved in different relations: this way, the aforementioned painting entities would be similar according to the relation *created by*, but different according to another relation. It does so by using an hyperplane, on which the entity representations for h and t are first projected before computing the function score as defined before.
Another implementation of this idea is TransR, proposed in Y. Lin et al. (2015). TransR is similar to TransH, but it introduces relation-specific spaces rather than hyperplanes, so that each relation is associated with a specific space in which the entity representations are projected before the score function is implemented.

Semantic-matching-based models

While translation models are based on computing distance between entities, models based on semantic matching make use of similarity-based scoring functions, that is the plausibility of facts is computed on the basis of the similarity of entities and relations according to their vector representations. One of the most impor-

tant models based on semantic matching in RESCAL, proposed by Nickel et al. (2011).
In RESCAL, entities are represented by vectors x, while relations are represented by matrices W. The score for a specific triple (h,r,t) is computed as $s = x_s^T W_h x_t$, and the representations are learned by comparing correct triples with incorrect ones through max-margin loss function.

Neural methods

Recently, as with natural language representation, neural models have been implemented in graph representation as well to obtain optimal entity and relation representations. Just as neural word embeddings are learnt by extracting information from contextual words (see Section 2.3), in neural KGE nodes' representations are computed through propagation of some notion of topological similarity / proximity (Huang et al., 2021). Since the similarities with language modeling (see Section 1.1), some neural models for KGE are based on models implemented in the field of NLP.

Graph walks A well-established methodology for the representation of graphs in the form of low-dimension vectors is through the use of *random-walks*, that is the process of starting at some vertex, and then at each time moving to another random, connected vertex. The random probability of traversing from one vertex to the other can be biased by the weight of the relations (in case the graph is weighted). The final sequences of randomly traversed nodes (called random walk) can be used to describe the graphs from which they were extracted (Lovász, 1993), and as such can be used as inputs for an embedding model to represent the graph as a whole.
One of the earliest attempts at implementing random-walk-based models for KGE is DeepWalk, proposed by Perozzi et al. (2014). The random paths generated are used as input to a generalization of the SkipGram algorithm (Mikolov, Yih, et al., 2013), that tries to predict the nodes present in the path (just like the original SkipGram tries to predict words in a sentence), and attempts at maximizing the probability of finding the correct node while minimizing the probability of finding an incorrect one. In doing so, the model takes into account both structural information (i.e., syntactical) and relational information(i.e., semantic) regarding the

nodes present in the graph.
As an improvement to DeepWalk, Grover et al. (2016) propose node2vec, in which the random walks are biased in order to balance between a breadth-first (BFS) traversal and a depth-first (DFS) traversal. By making this bias adjustable, this algorithm makes it possible to interpolate between retrieving information about community structure as well as structural equivalence (Goyal et al., 2018).
Ristoski et al. (2016) propose RDF2vec as a way to adapt the random-walk approaches to LOD data. The sequence used as the input of the embedding model is created by making use of two different approaches: graph walks and Weisfeiler-Lehman Subtree RDF graph kernels (de Vries, 2013). While the graph walks are generated by retrieving all the graph walks based on a starting root node and a given depth and iterating over the starting results, the RDF graph kernels are generated by retrieving the subgraphs that can be extracted from a starting root vertex and a given depth, then iterating over the results and combining all the results in a unique sequences. This path representation is then fed to the two algorithms presented in the word2vec model: Continuous Bag-of-Words and Skip-Gram.
RW-LMLM (C. Wang et al., 2019) is a more recently proposed model based on DeepWalk, which makes use of a language-model inspired objective. But in this case, the model also takes into account the order in which the nodes appear in the path, thus adding a further layer of information representation to the model.

Graph Convolutional Networks Kipf et al. (2016) propose a model for KGE based on Graph Convolutional Networks (GCN), which makes direct use of adjacent nodes' information to encode node embeddings. The basic GCN model can only capture information from immediately adjacent nodes, but additional layers on convolution can be added to access information from farther nodes. More formally, each node is initialized as a vector representing its features (for instance, a node representing a document could be represented by the embedding representation of its title). From this starting information, the node representation is then computed as $h_v = ReLU(\sum_{u \in N(v)} W x_u + b))$, where W and b are, respectively, learnable parameters for the weight matrix and the bias; $N(v)$ are the neighbors of node v, and $ReLU$ is the rectified linear unit activation function. On top of this, the original implementation in Kipf et al. (2016) implements normalization factors, not maintained in other implementations, such as the Syntactic-GCN proposed in Marcheggiani et al. (2017) (see Section 2.4).
By stacking the result of this formula k times, we encode information of k-farthest

nodes, thus improving encoding by including information coming from all over the graph, with Marcheggiani et al. (2017) showing that performance improves when reaching $k = 3$, but that further layers become largely redundant .

Finally, just as attention has become a central topic of discussion in natural language representation (Vaswani et al., 2017, and also Section 2.3), it has also been included in the processing and computation of KGE through the Graph attention network (GAT) model proposed by Veličković et al. (2017). The idea behind attention in this case is that the processing of node embedding should take into account the importance of information of other nodes, which can be more or less important according to the graph itself. Furthermore, the authors propose a node masking through which only adjacent nodes N are taken into account when computing attention, such that the structure of the graph is still taken into account. Attention, expressed as α, is included in the final computation of node embedding h_i' given the starting representation h_j through the formula

$$h_i = \delta(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in N_i}\alpha_{ij}^{k}W^{k}h_j)$$

2.4.4 Language-Aware Knowledge Graphs

The models for neuro-symbolic integration presented thus far mainly dealt with the injection of semantic knowledge from graphs into word embeddings. One of the issues behind these models is, thus, that the core source of information remains the Language Model, while leaving the information from a background Knowledge Graph as additional and optional data. Another possible approach is to encode linguistic information in the form of pre-trained word embeddings, store them as features for nodes/entities and then learn a Knowledge Graph Embedding (see Section 2.4.3) on top of it.

The main reasoning behind such neural-symbolic integration is that, while Knowledge Graphs are extremely useful in encoding grounded knowledge about words and concepts, they generally lack information about the context of use of said words, which is instead inherently encoded by distributional representations. Interest in this area has also been highlighted in the area of the Semantic Web by Hitzler et al. (2019).

In Baumgartner et al. (2018), for instance, the authors propose KADE, a model to align embeddings computed by two different models (e.g., a corpus and a background Knowledge Base) so that they can be represented on the same vector space. The main idea behind such a framework is that the information coming from textual corpora and those coming from a structured Knowledge Base can be considered complementary, and thus by representing them on the same vector space they could be combined.

One of the main areas of application of this framework is the integration of word embeddings in graph-based syntactic representation. Generally, while the idea is that linguistic information is inherently present in distributional representations, many researchers have shown the benefits that the integration of explicit linguistic information (e.g., dependencies or information about part-of-speech) might have on neural models. Works have proven the benefits of this approach for different tasks.

One such model is the one proposed by Al-Ghezi et al. (2020), but rather than using separate dependency trees for every sentence, they compose a single supergraph by unionizing all the sentence trees in the training dataset, in which sentences with similar grammatical structures (e.g., interrogative sentences) will be clustered together. In the final supergraph, words appearing in similar syntactic functions will appear together, and at the same time the global syntactic hierarchy of the training corpus will be preserved. The node2vec algorithm (see Section 2.4.3) is then used to learn node embeddings of the supergraph.

In particular, the information from a syntax tree built over a sentence is used to guide a series of syntax-based sub-networks which may represent several features, such as the parent-child relation or the sibling relation. While the system shows good results, it being trained from scratch makes its implementation consuming both in term of time and hardware usage.

An extensive list of works makes use of Graph Convolutional Networks (GCN) (Kipf et al., 2016) to integrate contextual word embeddings together with graph-based information. This was proposed by Marcheggiani et al. (2017) for the task of Semantic Role Labeling, with a dependency graph being used as an input for GCN. To enrich the graph with contextual representation of words, the authors initialize the token nodes using the output of a BiLSTM encoder. This framework was further investigated in Bastings et al. (2017) and Marcheggiani et al. (2018)

A similar model is also explored by Y. Zhang et al. (2018) for the task of semantic relation extraction. Using the results from the GCN, relations are then encoded through a concatenation of: sentence representation, subject entity representation and object entity representation.

Vashishth et al. (2019) explore the application of a GCN model to integrate further semantic information, in particular in the form of hypernymy, hyponymy and synonymy relations. The authors then evaluate the method on several general tasks such as word similarity, word analogy and text-based question answering.
As an extension to the GCN model is the Attention-guided GCN (AGGCN) proposed in Guo et al. (2019), which makes use of an additional layer that encodes multi-head attention attention (Vaswani et al., 2017) on top of the basic dependency tree.

Part II

Contributions

Chapter 3

Ontology-based Semantic Enrichment

3.1 Task Definition

The great majority of data available on the web is in raw text form, which is in stark contrast with the steadily increasing need for annotated datasets to develop and train Machine Learning (ML) systems to solve downstream and domain-specific tasks, as observed in efforts such as Leitner et al. (2020), Luz de Araujo et al. (2018), Usami et al. (2011), and Y. Wu et al. (2015).
In this regard, the enrichment of raw texts is not only useful in terms of ML applications, but it can be used to integrate semantic information from an external source of knowledge together with contextual information from word embeddings, particularly by making use of graph-based representations.
The standard models for semantic enrichment generally exploit the results from a Named Entities Recognition (NER) module in order to semantically enrich raw texts through the annotation of spans of text referring to proper names.
Despite the importance of NER on downstream tasks such as Information Extraction, Question Answering, Data Mining and more (Benikova et al., 2014), the results of such a module suffer from two main drawbacks that hinder their use for domain-specific applications.

First of all, as far as domain-specific texts are concerned, the aim of detecting proper name spans in a text should be extended to the recognition and classification of other linguistic realizations defining specific *concepts* of a specific domain knowledge. These concepts might represent important pieces of information that would be lost if only a NER module were to be used, such as *ultimi disegni di Leonardo* (Leonardo's last drawings), *fibula a trifoglio* (trefoil brooch) or *marmo pario* (parian marble).
Secondly, an additional issue arises in relation to the granularity of classification for the retrieved entities. As far as general-domain NER goes, these models generally implement a three-way classification of concepts, concerning PERsons, LOCations and ORGanizations. While these labels usually are enough for general-domain texts, domain-specific documents require a more fine-grained classification in order to capture and correctly identify the different types of relevant concepts. Brandsen et al. (2020) describe such differences in concepts classification for the archaeology domain in great detail.
This chapter describes the efforts made regarding the use of domain-specific ontologies and knowledge graphs in order to semantically enrich documents with domain-specific concepts, which have been previously proposed in the course of two separate works, namely:

- The creation of the NEAT dataset, as described in di Buono et al. (2023); and
- the implementation of ontological information on top of Multi-Word Expressions (MWEs) discovery, as described in Nolano et al. (2022a, 2022b).

3.2 NEAT - Named Entitites in Archaeological Texts

This section describes the efforts made in developing the Named Entity recognition in Archaeological Texts (**NEAT**) dataset[1], which won an award at the EuropeanaTech Challenge for AI/ML datasets[2]. The publication di Buono et al. (2023) describes the development of this resource in detail.
The proposed methodology aims at annotating unstructured texts with regards to domain-specific knowledge, as to provide a reliable source of training data for the task of NER in a domain- and language-specific environment through the imple-

[1]The dataset is freely available at https://github.com/unior-nlp-research-group/NEAT

[2]https://pro.europeana.eu/post/europeanatech-challenge-for-europeana-ai-ml-datasets-announcing-the-winners

mentation of syntactic and semantic information from several structured sources. An ontology is used on top of these methods in order to label entities with their correct domain-specific type.
While the presented methodology can be implemented in any domain (given that an ontology is available), this work focuses on the domain of archaeology and Italian language by testing the system on items' descriptions from the Europeana Collection[3]. Domain-specific terminologies and conceptual schema are used on top of the results to integrate domain knowledge.

3.2.1 Methodology

As previously mentioned, this methodology aims at obtaining a fine-grained, domain-specific entity classification, with a particular focus on terminological compounds. The main objective of the proposed methodology is to provide a consistent conceptual representation for domain-specific terms, and to exploit existing linked resources in order to improve consistency and reusability of the final dataset.

Off-the-shelf tools

Off-the-shelf, general-domain NLP modules can be used as a first step to incorporate additional information on top of an unstructured text. Such modules can provide several levels of information integration, and are generally easy to implement for many languages.
In particular, the tool used for the development of the NEAT dataset is the Italian NLP Tool (TINT)[4] (Palmero Aprosio et al., 2018), a standard NLP tool for the Italian language based on StanfordCoreNLP[5] which has been used to extract dependency graphs and named entities.
While dependency graphs are used to identify terminological compounds according to the methodology described below, the outputs of the NER module represent on their own semantic information. In particular, the module extracts spans of texts referring to people, locations or organizations and labels them accordingly. While these results from this first off-the-shelf (**OTS**) step are not fine-grained enough for the desired results, they can be easily harmonized with respect to a

[3]https://www.europeana.eu/en/collections
[4]https://dh.fbk.eu/research/tint/
[5]https://stanfordnlp.github.io/CoreNLP/

domain-specific ontology as described below.

Terminology Integration

The integration of domain-specific terminologies has been proven to be a valuable step in developing NLP and ML datasets (Peñas et al., 2001). The importance of such terminologies is proven by the many initiatives created to build common ground resources across languages and domains, such as the InterActive Terminology for Europe[6] (IATE) project by the Terminology Coordination Unit of the European Parliament[7] (TermCoord), the National Cancer Institute Metathesaurus[8] (NCIm), or the thesaurus for the Cyber Security Observatory of the CNR Insitute of Informatics and Telematics[9].
In order to integrate archaeology-specific knowledge, the ICCD thesaurus[10] represents a useful source of information. Furthermore, since it has been partially formalized into RDF format (Felicetti et al., 2015), its implementation can follow LOD principles. This resource[11] encompasses 4,048 entries and 31,631 relations, while also presenting fine-grained information (e.g., definition and editorial note), including a reference to the Getty AAT conceptual taxonomy[12] for 1,059 entries. In this work, the knowledge from the ICCD thesaurus is integrated into the unstructured texts through a step of Terminology Integration (**TI**).

Semantic Harmonization

The resources used for the two previously mentioned steps (the ICCD terminology and the off-the-shelf NLP module) present some issues regarding the conceptual representation of domain-specific terms.
As previously mentioned, 1,059 of the total entries in the ICCD thesaurus are

[6]https://iate.europa.eu
[7]https://termcoord.eu/
[8]https://ncim.nci.nih.gov
[9]https://iate.europa.eu
[10]http://www.iccd.beniculturali.it/it/ricercanormative/139/thesaurus-per-la-definizione-dei-beni-storici-artistici
[11]https://github.com/ICCD-MiBACT/Standard-catalografici/blob/master/schede-di-catalogo/ICCD$_T$hesaurus\%20$_c$ategorie$_s$ettori\%20disciplinari$_t$ipologie\%20di\%20schede\%20di\%20catalogo.rdf
[12]https://www.getty.edu/research/tools/vocabularies/aat/

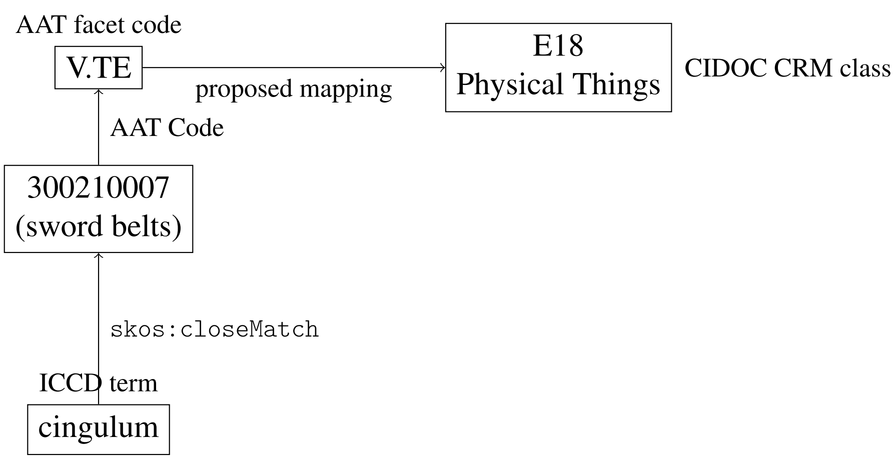


Figure 6: Example of semantic harmonization from an ICCD term to a CIDOC CRM class

linked to their corresponding classification code according to the Getty AAT taxonomy through the `skos:closeMatch` field. In particular, this taxonomy is organized by means of hierarchies and facets, representing a flexible and expressive representation schema (Denton, 2003). Nevertheless, this framework is intended to describe a general classification for art and architecture, and it cannot be considered representative enough for classifying entities in the narrower archaeological domain. On top of this, Denton (2003) highlights three major issues connected to facet classification: difficulty of choosing right facets, the lack of ability to express relationships between them, and the difficulty for general visualization.
Similarly, with regards to the NLP module, the results are not fine-grained enough for a domain-specific representation, since they only cover people, locations and organizations.
As a better option, the CIDOC Conceptual Reference Model[13] (CRM) can be used as a domain-specific representation schema for the retrieved entities (Doerr, 2003). In order to do so, both results from terminology integration and off-the-shelf NER are aligned to classes from the CIDOC CRM, according to manual mappings.
An example of such a process is shown in Figure 6, and Section 3.2.2 describes the harmonization process in detail.

[13]https://www.cidoc-crm.org/

Semantic Projection

Traditionally, the task of semantic projection aims at exploiting semantic information extracted from one richer resource to infer structures in a poorer target. Examples of semantic projection can be found in David et al. (2001), Fung et al. (2004), Modi et al. (2012), Padó et al. (2005), and Ribeiro et al. (2020).
In this work, a step of Semantic Projection (SP) is implemented to project domain-specific linguistic occurrences in the form of terminological compounds from a manually annotated corpus onto non-annotated examples. This process is based on the assumption that there exists knowledge patterns in the form of predictable and recurring patterns which are used to express a given conceptual representation (Meyer, 2001).
Following Faber et al., 2014, such conceptual patterns are also assumed to connect lexical units or terms belonging to the same part of speech and having the same syntactic distribution (i.e. they can fulfill the same syntactic functions).
Thus, phenomena such as hyponymy and hypernymy, that connect a general term to a more specific one as they share the same semantic content, are generally represented by recurring patterns that can be predicted, in the form of terminological compounds.
Such relations, generally represented in a taxonomy, define increasing specificity according to some conceptual features, such as use or shape. For instance, the term *bracciale* (bracelet) represents a domain-specific term which can be further specified according to use in the terminological compound *bracciale omerale* (humeral bracelet) and according to shape in the compound *bracciale a spirale* (spiral bracelet). Similarly to single terms, these compounds define a concept and its characteristics, and as such should be classified as a full entity.
In order to recognize and classify such terminological compounds, a domain-specific, annotated set of texts is annotated using a subset of the CIDOC CRM. Then, starting from the semantic and syntactic heads of the annotated spans of text, we infer syntactic and semantic patterns that are then projected onto an unstructured text in order to retrieve previously unseen compounds.

3.2.2 NEAT Dataset

In order to prove the validity of the presented approach and investigate possible future improvements, the methodology hereby described is then used to create the NEAT dataset. In particular, the final dataset has to provide the following levels of information:

- PoS information;
- named entities annotation, including nominal compound annotation;
- entity-level concept description.

The following sections describe the steps taken in order to achieve these objectives, and make considerations regarding the processes involved.

Data Collection

As previously mentioned, the methodology is tested on a specific domain and a specific language. In particular, the focus is on the domain of archaeology and on Italian language, thus restricting desired textual contents. Furthermore, the methodology is to be implemented on real-life environments, and as such texts are to come from actual cultural heritage providers.

The Europeana respository[14] represents the source of such texts, since it provides digital access to European cultural heritage from actual cultural institutions. The data is thus both trustworthy and authentic, and aggregated under Thematic Collections in relation to either topics (e.g., Archaeology and Sport), features (e.g., Chinese Heritage and Pandemics) or custom galleries (e.g., Musical Miniatures and Irish Ballads). For each cultural item, textual description is also provided in one or more languages.

The entries of the Europeana Repository are further enriched through the Europeana Data Model[15] (EDM), which connects the data using a coherent metadata schema composed of several namespaces (such as RDF, RDFS and SKOS).

The data is accessed through the Europeana Search API[16], through which unstructured textual descriptions about cultural heritage items can be retrieved.

More specifically, the data is filtered according to domain-specific classes. In particular, the ones taken into account are `europeana_aggregation_edm_country` (defining the country of provenance of the data) and `collection` (describing the collection they belong to). The data is filtered based on the country of provenance rather than the language of the description to be sure that the texts are originally written in Italian, and not translated from another language. The query[17], is thus

[14] https://www.europeana.eu/

[15] https://pro.europeana.eu/page/edm-documentation

[16] https://pro.europeana.eu/page/search

[17] https://api.europeana.eu/record/v2/search.json?wskey=APIKEY&query=europeana\$_a$ggregation$_e$dm\$_c$ountry:Italy+AND+collection:archaeology&rows=100&cursor=*

set so that the former class is equal to `Italy` and the latter to `archaeology`.
For each entry the following values are retrieved:

- `dcTitle`, the title of the item described by the entry;
- `dcDescriptionLangAware`, a key:value dictionary where the keys are represented by language tags, each paired with a textual description in that specific language.

Values are extracted for a total of 5,000 items, 1,000 of which are used for an evaluation step of the process.

Off-the-shelf tools

Starting from the retrieved texts, a preprocessing step is performed using off-the-shelf NLP tools. In particular, this is done using the Italian NLP Tool (TINT)[18] (Palmero Aprosio et al., 2018), a standard NLP tool for the Italian language based on StanfordCoreNLP[19]. The following data is retrieved from the outputs of this tool: the dependency and POS representations of each sentence, and the results from the NER module.
While the dependency representations are not explicitly present in the final annotated corpus, this information is used to identify terminological compounds, as described in Section 3.2.2. Similarly, the POS module is used to remove some of the noise from the results in the Evaluation step.
The implementation of this kind of linguistic information greatly improves the final results, but it also presents some issues that are described in Section 3.2.2.
The output of the NER module, on the other hand, is used as the first step of terminology annotation of the dataset, namely OffTheShelf (OTS).
In order to make the results coherent with more domain-specific sources of knowledge, the original PER, LOC and ORG labels are mapped to their corresponding label according to the CIDOC Conceptual Reference Model, according to the mapping described in Section 3.2.2.

Terminology Integration

The second step in the proposed methodology makes use of grounded terminologies to incorporate domain-specific terms in the extracted texts. In particular, the

[18]https://dh.fbk.eu/research/tint/
[19]https://stanfordnlp.github.io/CoreNLP/

open data version of the ICCD thesaurus is used to enrich the texts at the level of tokens.
In the thesaurus, each term is characterized by a `skos:prefLabel` property, describing its textual form, and by a `rdfs:label` property, which defines the specifications of the head with regard to its conceptual representation. For instance, an entry might have *statua maschile* (male statue) as its `skos:prefLabel`, and *maschile* (male) as its `rdfs:label`, since it specifies the head *statua* (statue). The values for both properties have their own `uri`.
Furthermore, each entry is linked to the Getty AAT through a `skos:closeMatch` property. By exploiting this value, it is possible to extract for each term its value according to the AAT conceptual hierarchical structure, in the form of its facet, represented by a specific code. These codes are then mapped to their corresponding classes in the CIDOC CRM, according to the process described in Section 3.2.2.
Starting from the list of terms in the thesaurus, a simple exact match search is implemented in order to first recognize any entity occurring in the texts, and then classify them according to its entity type.
While some results are achieved from this initial step, several entities are only partially annotated using an exact match search. For instance, in the sentence:

la scultrice modella <u>*busto*</u> *di creta raffigurante Koblet*

(the sculptress molds the clay <u>*bust*</u> *depicting Koblet)*

while the word *busto* (bust) is correctly recognized by the system, it can only be considered partially correct, since the full span *busto di creta raffigurante Koblet* (clay bust depicting Koblet) should be retrieved. Since this compound is never seen in the thesaurus, a further step of Semantic Projection is needed in order to expand on these initial results, as described in Section 3.2.2.

Semantic Harmonization

To obtain a more fine-grained annotation suitable for the representation of conceptual references in the specialized domain of archaeology, a mapping process is implemented, so that the AAT codes previously extracted can be mapped over the subset of selected classes from the CIDOC CRM. Table 3 shows the manually-defined mappings implemented in the course of the work. A similar process is followed in order to harmonize the results from the off-the-shelf NER module with the CIDOC CRM classes. This process is more trivial, since the three la-

AAT facets	CIDOC CRM classes	Occurrences
B.BM - Associated concepts	E28 - Conceptual Thing	4
D.DC - Attributes and Properties	E54 - Dimension	3
D.DG - Design Elements	E18 - Physical Thing	7
K.KD - Disciplines	E41 - Appellation	1
K.KT - Processes and Techniques	E41 - Appellation	9
M.MT - Materials	E57 - Material	34
V.PE - Object Genres	E18 - Physical Thing	69
V.PJ - Components	E18 - Physical Thing	162
V.RK - Single Built Works	E18 - Physical Thing	25
V.T - Furnishings and Equipment	E18 - Physical Thing	666
V.VC - Visual Works	E18 - Physical Thing	56

Table 3: Mapping of AAT facets and CIDOC CRM classes

bels employed for the NER tool (LOC, PER and ORG) can be easily mapped to archaeological-specific classes, as defined in Table 4. One important issue in

NER labels	CIDOC-CRM classes	Occurrences
PER	E39 - actor	5427
ORG	E39 - actor	2216
LOC	E53 - place	3762

Table 4: Mapping from NER labels to CIDOC CRM classes

this conversion process is the differences in the conceptualization of the domain implemented in the different resources. For instance, the AAT applies a fine-grained representation to classify people and organizations, i.e. `H.HG - People` and `H.HN - Organizations`, while the CIDOC CRM represents a unique class for both entities, namely `E39 - Actor`.

Semantic Projection

Finally, a step of semantic projection (SP) is implemented, in order to enrich the texts in relation to the CH domain, by using a corpus of Italian archaeological texts annotated with a subgroup of the CIDOC CRM ontology[20].

[20]The efforts to build this corpus go behind the scope of the present work, but are explained in detail in di Buono et al. (2023).

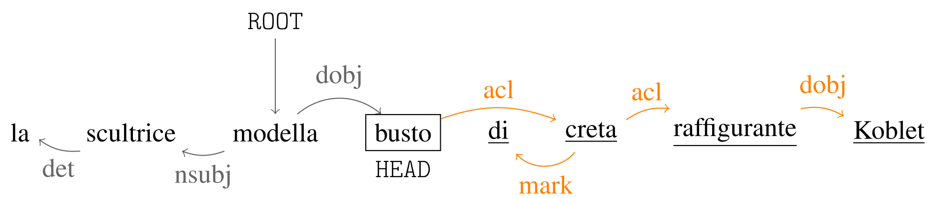


Figure 7: Example of the Semantic Projection step

An initial step of exact match search based on the terms extracted from this annotation is first tested. As previously shown in Section 3.2.2, this exact match search suffers from low precision, given the high number of unseen terms present in the unstructured texts. In order to solve this issue, an algorithm is devised, expanding on the work by di Buono (2017), to extract such previously unseen domain-specific compounds from the texts, which follows the steps hereby described.

1. once a single-word term (either extracted from the ICCD thesaurus or the annotated corpus) is found in one of the items' descriptions, it is regarded as the head of a potential terminological compound
2. any word in a window of 5 tokens preceding and following the term that is governed by it according to the dependency representation is used to populate a list of syntactic dependencies
3. this list of syntactic dependency is further iteratively populated by adding any dependent of the words already present in the list
4. the process continues until a punctuation mark is reached, or until no new dependencies can be extracted

An example is shown in Figure 7, where the entity *busto di creta raffigurante Koblet* (clay bust depicting Koblet) is extracted starting from the term *busto* (bust), considered the HEAD of the candidate compound.

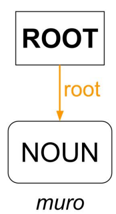
Nevertheless, this process does not always retrieve terminological compounds, as it is based on dependency representations and does not make full use of the linguistic information that can be extracted from domain-specific realizations.

In order to improve the result, a grounded set of terms is identified from the annotated corpus, and then used to recollect all the known knowledge patterns. These patterns can then be used on the initially extracted compound candidates to filter out anything that should not be considered part of a conceptual realization.

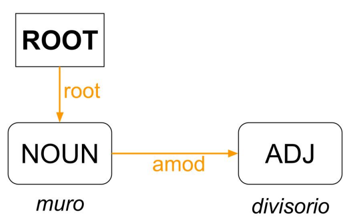
More formally, given a proposed terminological compound extracted from a de-

pendency list, its components are filtered using the known knowledge patterns spanning from the same head. In case its structure adheres (at least partially) to any of the possible patterns for its head, the subgraph adhering to the knowledge pattern can be considered an entity. In case it adheres to more than one structure, the longest one is taken into consideration.
In particular, these knowledge patterns are extracted in the form of their dependency path and their POS. For instance, given the term *muro* (wall) and the terminological compounds in which it appears as syntactic head, the following patterns are extracted[21]:

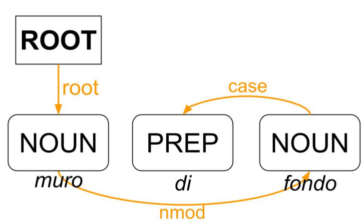
- *muro* (wall)



- *muro divisorio* (dividing wall)



- *muro di fondo* (back wall)



Future unseen terminological compounds stemming from the HEAD *muro* are assumed to follow at least one of these knowledge patterns in terms of POS and dependency structure. For example, given the sentence:

[21] The dependency graphs are extracted by means of TINT, https://dh.fbk.eu/research/tint/

Si trovava oltre il muro di pietra del forte italiano.

(It was found behind the stone wall of the Italian stronghold)
and the extracted term *muro* (wall), first the full dependency path spanning from it is extracted, as shown in Figure 3.2.2.

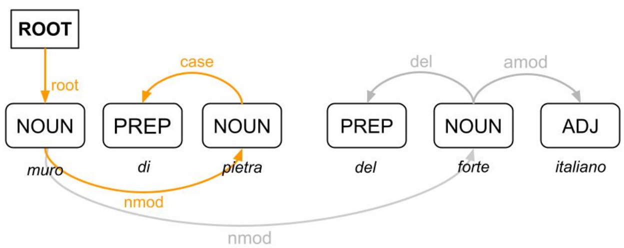


Figure 8: Dependency parsing of the candidate term "muro di pietra del forte italiano"

The pattern highlighted in orange in the figure, which describes the compound *muro di pietra* (stone wall) is also found in the annotated span *muro di fondo*, and should thus be considered a terminological compound. The span of text *del forte italiano* (of the Italian stronghold) can be left out as it is not part of any knowledge pattern for this specific term.

Evaluation

While the process shows interesting results, drawbacks are still present, mainly caused by its reliance on automatic modules and ontologies that can hinder its performance.
First of all, inconsistencies arise with regards to the syntactic parser: some tokens get assigned the wrong POS tag (which directly hinders the retrieval of terminological patterns), inconsitencies can arise due to language-specific phenomena, such as the agglutination of clitic particles in the form VERB+CLITIC[22], and inconsistencies finally arise as caused by the presence of less-frequent, domain-specific terms, as also observed by Coden et al. (2005a) and Mikheev (1997b).

[22]It is for instance the case of the word *pentola* (pot) which is misinterpreted as the verb *(mi) *pento* (I repent) + the clitic *la* (her).

Furthermore, a specific step of **entity evaluation** is manually performed by three expert Italian linguists on the validation section of the dataset. In particular, results are evaluated in terms of acceptability of retrieved text spans and classified entity types.
The first part of the evaluation takes into account the criterion of informativeness of the text, by evaluating if the extracted text spans contain all the elements which allow to get valuable information about the head of the compound. The criteria are the following:

- the text should not present elements referring to other elements not explicitly mentioned in the noun phrase;
- the text should represent a single intelligible noun phrase on terms of meaning;
- the text should be a complete noun phrase.

The second step of the evaluation is interested in whether the texts were correctly labeled according to the CIDOC CRM classes. Table 5 shows the outcomes of the evaluation performed on the NEAT TI and NEAT SP datasets. With regards to the

Step	**Candidates**	**Valid Text (%)**	**Valid Entity (%)**
TI	609	457 (75%)	413 (90%)
SP	2,554	1861 (72%)	1455 (78%)

Table 5: Candidate Evaluation

TI step, the methodology proposed returns less results, but it allows the identification of more precise possible entity candidates, as shown by the 90% percentage of valid entities. During the SP step, on the other hand, way more results are obtained, but the procedure is less precise in labeling the entities.
The problems emerged during the evaluation of the TI and SP steps are similar. First of all, issues arose in relation to the classes `E53 - Place` and `E18 - Physical Thing`: in several instances these two classes are confused. It is often the case that `E53` entities are classified as `E18` instead. This is partly due to some ambiguity inherently present in the CIDOC CRM classes in relation to specific terms: *chies* (church), for instance, can both be interpreted as a "persistent physical item with a relatively stable form, human made" (`E18`) and an "extent in space, in particular on the surface of the earth" (`E53`).
Similarly, entities referring to `E5 -Event` are often confused with the `E18` class.
Finally, some problems are related to the extraction of long spans of texts, and entities with different classes co-occurring in the same span of text.

3.2.3 Final Results

Once the proposed methodology has been tested and evaluated on the 1000-texts mock dataset, it is applied on a final dataset composed of the Italian descriptions for 5,000 items extracted from the Europeana Collection. The final proposed result is an annotated dataset, providing a reliable source of data for training domain-specific models for the domain of archaeology.
This final dataset is composed of a total of 349,354 tokens annotated with a IOB-formatted annotation (Ramshaw et al., 1999) specific to the CIDOC CRM ontology. Table 3.2.3 shows the total number of annotated spans of texts.

Type	OTS (unique)	TI (unique)	SP (unique)	Total
E2 - Temporal	N/A	-	-	-
E4 - Period	N/A	-	94 (12)	94 (12)
E5 - Event	N/A	-	385 (90)	385 (90)
E18 - Physical Thing	N/A	2,228 (630)	15,482 (3,610)	17,656 (4240)
E28 - Conceptual Thing	N/A	-	233 (68)	233 (68)
E39 - Actor	7,643 (1,261)	-	8,011 (1,365)	15,654 (2,626)
E41 - Appellation	N/A	-	111 (35)	111 (35)
E52 - Time Span	N/A	-	-	-
E53 - Place	3,762 (400)	-	886 (195)	4,648 (595)
E54 - Dimension	N/A	2 (1)	2 (1)	4 (2)
E57 - Material	N/A	267 (54)	1,950 (83)	2,217 (137)
Total	**11,405 (,1661)**	**2,497 (685)**	**27,100 (5,459)**	**41,002 (7,805)**

Table 6: NEAT Entities

The majority of the extracted results comes from the annotated corpus, making up for the 66.09% of the total annotations. This is mainly caused by the originally annotated corpus covering not only named entities but also general archaeological concepts. The results from the TI phase, on the other hand, represent the lowest percentage of the total annotated entities, covering up the 6.08% of the total. Despite the low number of results, as noticed in Section 3.2.2, these retrieved entities tend to be generally precise, albeit very strict.
Finally, the OTS results, while not being specific to the archaeological domain, represent a solid baseline to annotate general-domain entities which can still be considered useful for our task. In particular, OTS is especially useful in annotating places (`E53 - Place`), for instance compounds such as *via dei Fori Imperiali* (street of the Imperial Fora).

3.3 From Multiword Expression Discovery to Concept Enrichment

This section proposes a methodology for the semantic enrichment of unstructured texts, starting from automatically extracted multiword expressions (MWEs) and integrating semantic information using ontological knowledge.
This methodology was first proposed in the course of the Conference on the Future of Europe (COFE) Datathon[23], both for the task of keyword extraction (Nolano et al., 2022b) and concept clustering (Nolano et al., 2022c). An extension of this methodology has been applied to the domain of Cultural Heritage, as described in Nolano et al. (2022a).
The core idea is to extract candidate MWEs by expanding keywords automatically retrieved from the texts, then connecting these MWEs to the concept they most closely match given a background KG. This correlation is used to filter the MWEs so that the ones that most strongly correlate to domain-specific concepts are kept, while others are discarded.
The following sections describe the methodology in detail, the experiment made and finally provide an analysis of the results obtained.

3.3.1 Methodology

The proposed methodology makes use of extracted MWEs and interconnected RDF data. More specifically, MWE candidates are first extracted from unstructured texts and then connected to the concepts they are strongly related to, in the form of specific RDF items. The steps performed are the following:

1. MWE discovery
2. Concept Discovery
3. Ontology-based filtering

Figure 9 shows the first two steps in graphical form, while Figure 10 shows an example of a graph used for ontology-based filtering.

[23] https://futureu.europa.eu/en/pages/datathon

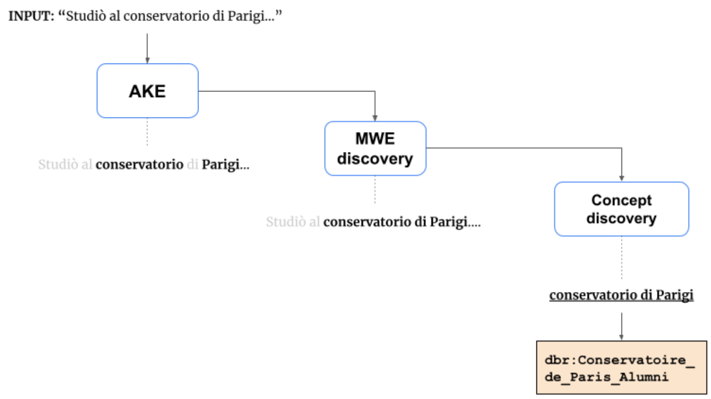


Figure 9: Graphical representation of the process of monolingual MWE discovery: in the first step, Keywords are automatically extracted using the pke Python library. The automatically extracted keywords are then expanded based on specific patterns. The extracted MWEs are connected to the link they most likely refer to, through similarity measure with the italian label of said link.

Monolingual MWE Discovery

The starting point of the proposed methodology is the retrieval of domain-specific terms from raw texts in the form of keyphrases. In particular, such keywords are extracted using off-the-shelf libraries. In order not to put any restriction on the type of semantic information to extract, an automatic keyphrase extraction (AKE) module is used to retrieve keyword candidates from the text. While these modules extract keyphrases using no explicit domain-specific information, they make use of statistical and structural information about the text to retrieve meaningful results. As previously mentioned (Section 2.4.2), such models benefit from the implementation of graph-based text representations, which are shown to greatly increase their results.
While the results obtained through these modules can be considered useful, they do not cover the full amount of knowledge present in texts. In particular, while there is no restriction on the type of keywords that can be extracted, the modules generally fail to retrieve full compounds, and tend to extract only single-word terms. For instance, in a sentence such as *Ha studiato presso il Conservatorio*

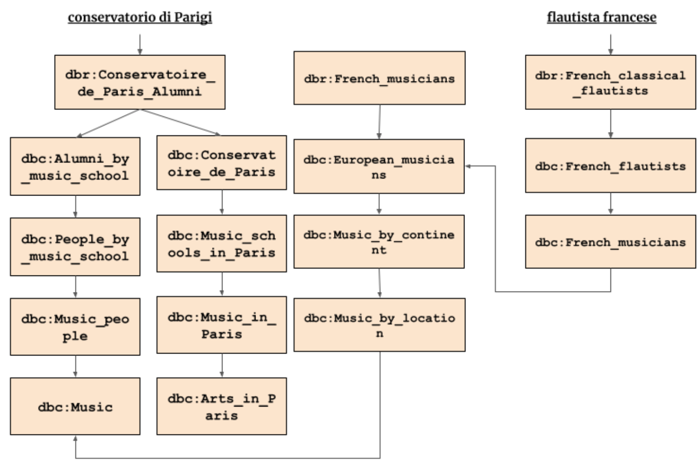


Figure 10: Graphical representation of the graph implemented for ontology-based filtering. The graph is constructed by traversing the `skos:broader` property 4 times from the links available for the entity. The nodes representing concepts related to the entity will generally share several edges, thus giving more weight to the MWEs linked to them.

di Parigi (he studied at the Paris Conservatory), the AKE module would extract only *Conservatorio* (Conservatory) and *Parigi* (Paris), without retrieving the full compound.

In order to solve this issue, the original keywords retrieved from such a module represent a first list of keyword candidates which are then used as input for an MWE discovery phase. More formally, after the keyphrase extraction phase, the extracted candidates are checked in context, and a candidate w_x is assumed to be part of an MWE in case it is close to another candidate w_y and the two candidates are separated by specific hand-defined linguistic patterns based on heuristics, listed in Table 7.

Such patterns are useful in expanding automatically extracted keyphrases to cover more domain-specific terms, while also being flexible enough to be used in a multilingual environment, as shown in Nolano et al. (2022b). For instance, given the sentence *Compliance and implementation of carbon prices is a solution to mitigate pollution levels*, the following keyphrases are extracted:

Distance	Pattern	Example
0	w_1 w_2	*Adelaide Festival* (Adelaide Festival)
1	w_1 *PREP* w_2	*nuova generazione* ***di*** *musicisti folk* (new generation of folk musicians)
	w_1 *ADJ* w_2	*Aleksandra Aleksandrovna* ***nata*** *Grigorovič* (Aleksandra Aleksandrovna nee Grigorovič)
2	w_1 *PREP DET* w_2	*premio Nobel* ***per la*** *letteratura* (Nobel prize for Literature)
	w_1 *ADJ PREP* w_2	*direttore* ***principale della*** *Philharmonia Orchestra* (main directory of the Philharmonic Orchestra)

Table 7: Candidate patterns

compliance and $\boxed{\text{implementation}}$ of $\boxed{\text{carbon prices}}$ is a solution to mitigate pollution level

$w_1 \longrightarrow \mathit{PREP} \longrightarrow w_2$

Since the two keywords are divided by an acceptable pattern, they are combined in a single compound *implementation of carbon prices*, which is more informative than the two keywords divided.
In Nolano et al. (2022a, 2022b) it was shown that all the defined patterns are productive, even though not to the same extent. In particular, in Nolano et al. (2022a) it is shown that, for the Italian language, the most productive pattern is w_1 PREP w_2, while the least productive one is w_1 ADJ w_2.
Once this step is completed, each extracted MWE is given a relevance score and classified according to the conceptual category assigned to its head token.

Concept Discovery

While the retrieved keyphrases already represent some sort of domain-specific knowledge extracted from the text, this representation is yet to be made explicit. As mentioned already, the explicitation of such knowledge is an important step in making such semantic information accessible by a machine. In particular, the objective is to represent such knowledge in a coherent and reusable way so that it can be implemented in more complex systems (such as QA modules and chatbots).
In particular, graph-based knowledge is implemented, in the form of linked open data, as a source for such semantic information. This knowledge can come from

different sources, for instance one such source is represented by the EuroVoc thesaurus[24] in Nolano et al. (2022b), and by DBpedia[25] in Nolano et al. (2022a). The proposed methodology is the same regardless of the knowledge base implemented: the extracted MWEs are scored against entries from the knowledge base, according to similarity scores between the MWE and entries' label, thus creating a network of connections between raw spans of text and entities.

Ontology-Based Filtering

The information collected in the previous step is used to filter extracted candidates, by reranking them according to their connection strength to domain-specific concepts. More specifically, all the links extracted are used to recreate a hierarchical graph by traversing the relation defining hyperonymy, such as the `skos:broader` relation for Dbpedia. For each entry extracted from the knowledge base, every node up to the 4th highest node in the hierarchy is retrieved.
The obtained graph shows a representation of interconnected concepts. From this graph, the concepts extracted needs to have to following features:

- they are connected to an extracted MWE
- they are strongly connected to other nodes in the graph

Thus, the graph is used to reweight domain-specific keyphrases and MWEs, while also representing an additional source of semantic information to enrich the data at hand.

3.3.2 Experiment

Data

While the methodology can be generally considered dataset agnostic, the implemented data affects what kind of background knowledge base can be used for the process of semantic enrichment.

For instance, in Nolano et al. (2022b) the texts had no explicit connection to a structured knowledge base, and as such the choice of which background knowledge source to use is particularly important. As mentioned already, in this work

[24]https://data.europa.eu/data/datasets/eurovoc?locale=en
[25]https://www.dbpedia.org/

the source of knoweldge is represented by the EuroVoc thesaurus, which comprises multilingual and multidisciplinary vocabularies covering the activities of the EU in 24 EU languages. In order to access structured data from EuroVoc, a REST API service[26] is used.
The entries of EuroVoc are classified according to a hierarchical taxonomy, which can be used to build a graph-based candidate representation, as defined in Section 3.3.2 and Section 3.3.2.

In Nolano et al. (2022a), on the other hand, the proposed methodology is implemented on domain-specific texts, namely CH-related entities' descriptions, collected using the Europeana Entity API[27].
These entities represent a collection of Named Entities harvested from several online data catalogs (e.g., Geonames, DBpedia, and Wikidata). For the purposes of this work, biographical information about entities of type `agents` are extracted, representing artists from different cultural heritage sub-domains such as music and fashion.
In particular, using the SPARQL Entity API the following information are retrieved:

- the English label
- the DBpedia entry they are linked to
- the Italian text for their biographical information

Given this information, the objective is to semantically enrich the Italian biographical descriptions. Since using the whole DBpedia as a background Knowledge Base would be particularly intensive, and might result in noisy outputs, the list of candidate concepts is limited to the list of links connected to the DBpedia entry representing the entity. These links can be accessed using the DBpedia SPARQL endpoint[28], in particular by traversing the `dbo:wikipageWikiLink` property for the entry.

MWE Discovery

Keyphrases are first extracted for each text by using the `pke` Python library[29], and using the MultiPartite Ranking algorithm (Boudin, 2018). This module outputs a list of keyphrases, each accompanied by an algorithm-specific relevance score,

[26] https://www.vocabularyserver.com/eurovoc/
[27] https://pro.europeana.eu/page/entity
[28] https://dbpedia.org/sparql
[29] https://github.com/boudinfl/pke

which generally ranges from 0 to 1. As mentioned already, from this list of automatically extracted keywords, new MWEs within the text are to be discovered. To do so, keyphrases are checked so that when they are close enough[30], and when the sequence of tokens between them is acceptable according to predefined patterns of co-occurring elements, they can be combined to form new keyphrases. Each candidate is checked in both w_1 or w_2 positions.
Each newly extracted MWE is assigned a new score by averaging the Multipartite Ranking value for each of the keyphrases involved in the MWE. Keyphrases which cannot be used to build any new MWE are kept as they are.

Concept Discovery

The next step is connecting the extracted MWEs to domain-specific concepts they refer to. In order to extract these concepts, background knowledge bases are implemented, as defined in Section 3.3.2.
In Nolano et al. (2022b) the top-3 closest matches for every keyword are extracted using the EuroVoc web API, and then each is stored together with their hierarchical structure.

In Nolano et al. (2022a), on the other hand, first all the hyperlinks present in the entity's corresponding Wikipedia page are extracted, as described in Section 3.3.2. These hyperlinks represent domain-specific concepts related to the original entity at hand. Then, the Italian labels for each of these hyperlinks is retrieved, One main issue with these labels is that their Italian form is not always present, as is the case for most of the category-defining hyperlinks such as `dbp:Victorian_poets` and `dbc:19th-century_English_poets`, for which only the English label is available. Since these links generally refer to domain-specific knowledge classification, they should be accessed even in absence of an Italian label. In order to do so, in case an Italian label is unavailable for a specific link, they were automatically translated from English using the Argos Translate Python library[31]. Once these Italian labels are retrieved, each MWE is connected to the concept it most closely matches, among those concepts present as hyperlinks for the specific entity. More specifically, two similarity measures are applied between the MWE and each of the concepts:

[30] In Nolano et al. (2022b) this window is set at three elements, in Nolano et al. (2022a) it is set at two elements.
[31] https://github.com/argosopentech/argos-translate

1. cosine similarity between vector-space representations[32],
2. overlap coefficient between the surface forms[33]

By implementing these similarity measures together, semantic similarity between vectors is taken into account, while also considering the cases in which the MWE and the label share similar surface form, despite their vectors being distant. In particular, the two similarity scores are combined by computing the raw product between them.

From this list of computed similarity scores, any $MWE, label$ pair with a similarity score lower than 0.4 is discarded. Then, for each remaining MWE, only the label with the highest similarity score is kept. In case a MWE is not linked to any of the available concepts, its score is left as is, according to the AKE module for the following steps.

Ontology-Based Filtering

The MWEs, together with the concept they most strongly refer to, are used to build a conceptual graph, which is then employed to rerank MWEs according to their connection strength in this graph.

The graph is built according to the hierarchical structure of the concepts retrieved in the previous step: the full hierarchical structure is extracted in Nolano et al. (2022b), while in Nolano et al. (2022a) this is done by traversing the `skos:broader` property 4 times. The main idea is that in this kind of conceptual graph, concepts that are related to each other will generally share connections to common nodes. In particular, the connection strengths of the concepts related to extracted MWEs is extracted, so that the MWEs strongly connected to central concepts can be reranked accordingly.

More specifically, the betweenness centrality (Freeman, 1977) is computed for each of the connected concept nodes. In Nolano et al. (2022b) this value is integrated together with the original multipartite ranking score, while in Nolano et al. (2022a) it is implemented together with the original multipartite ranking score and the two-ways similarity score. In both cases, the two values are simply added together.

Once the centrality scores are integrated, the MWEs are reranked according to their newly computed scores, and the highest-scoring ones are stored together

[32] Pre-trained fastText word vectors for Italian are implemented.

[33] The Szymkiewicz-Simpson coefficient, as described in M K et al. (2016), is implemented.

with semantic information about the concepts they refer to.

Chapter 4

Graph Neural Models for Relation Extraction

4.1 Task Definition

The task of relation extraction (**RE**) is the task of predicting a semantic relation connecting entities appearing in a given document (Bach et al., 2007). Formally, a relation is defined as a tuple $t = (e_1, e_2, ..., e_n)$, where each e_i represents an entity involved in the relation r, stored in a background ontology or knowledge base, given a document D.
While relations may happen between several entities, this task is generally used to retrieve binary relations (i.e., linking two and only two entities) (Bach et al., 2007).
For instance, given the sentence *Leonardo da Vinci painted the Mona Lisa*, with `Leonard da Vinci` and `Mona Lisa` annotated as entities, an RE model aims at predicting the relation $creator(LeonardodaVinci, MonaLisa)$. Such a relation can also be represented by the triple $(LeonardodaVinci, creator, MonaLisa)$, as expressed in a triple store.
Regardless of the representation implemented, the relation extraction module is asked to retrieve the `creator` label, or its corresponding label in a given background ontology. For instance, the same relation is expressed in Wikidata using

the form of `P170 - creator`.

This task is of extreme importance for many NLP tasks, such as question answering (M. Yu et al., 2017), knowledge base population (Y. Zhang, Zhong, et al., 2017) and knowledge discovery (Quirk et al., 2017).

As it is the case for many other NLP tasks, the use of distributional representations in the form of word embeddings trained using neural architectures is the go-to option to solve RE. Over the years, many such models have been proposed, from end-to-end systems (Miwa et al., 2016; M. Zhang et al., 2017) to fine-tuned models, using either static embeddings (P. Zhou et al., 2016b) or contextual embeddings (Baldini Soares et al., 2019).

It is of interest that RE is one of the main areas where the integration of external knowledge can be implemented on top of textual information. This is mainly due to the nature of the task itself, which directly leads to the integration external knowledge, in the form of information about relations (e.g., its range and domain constraints) and about entities (e.g., their types).

Similarly, the integration of syntactical information (in particular in the for of dependency graphs) has also been explored for this task as well, proving that such information is useful in disambiguating which semantic relation exists between two entity mentions (see Sections 2.3.3 and 2.4.4).

This chapter is dedicated to a preliminary analysis of graph-based models and semantic and syntactic knowledge integration for the task of RE. In particular, graph-based representations, implemented using GCN (Marcheggiani et al., 2017), are paired against general text-based models for RE in both a general environment and in an adversarial framework (Jia et al., 2017), in order to test the actual performance of knowledge integration, and the robustness of the models under pressure.

Acknowledgement The work described in this chapter was carried out in collaboration with Moritz Blum and under the supervision of Basil Ell and Philipp Cimiano (Semantic Computing Group, University of Bielefeld).

4.1.1 Data

In order to carry out specific experiments on the RE task, a dataset is specifically created. The starting point of such a dataset is FewRel[1] (Gao et al., 2019; Han et al., 2018), a large few-shot RE dataset created through a combination of distant

[1] https://github.com/thunlp/FewRel

```
{"P177":
  [
  {"tokens" :
    ["The", "Ohio", "Connecting", "Railroad", "Bridge", "crosses", "the", "Ohio", "River", "at", "the", "island", "."],
  "h" :
    ["ohio connecting railroad bridge", "Q7080785", [[1,2,3,4,]]],
  "t" :
    ["ohio river", "Q4915", [[7,8]]]
},
...
]
}
```

Figure 11: Example of entry from FewRel. Each sentence is grouped according to the relation they realize (**P177**), then represented by its **tokens**, its head entity (**h**) and its tail entity (**t**).

supervision and human annotation. While the original objective of FewRel is to train models on few-shot RE, with some tweaks it can also be used in a standard RE environment.

More formally, the original dataset is organized as follows: for each relation (expressed with a Wikidata label), a list of examples is stored, each tokenized and with head and tail entity mentions highlighted, together with their corresponding id on Wikidata. An example of a FewRel entry is shown in Figure 11.

In order to create the dataset implemented in this work, the `train_wiki` and `val_wiki` sections of FewRel are combined. These subsections contain, respectively, 64 and 16 relations, each with 700 examples, for a total of 80 relations and 56,000 sentences.

The combined dataset is randomly split into train/test/dev splits with percentages 70/15/15 so as to train predictive models. This dataset will be referred to as **FewRel (custom)**.

While not used in the adversarial part of the evaluation, two more datasets are taken into consideration to further evaluate proposed models. The first one of these datasets is **T-REx (custom)**, a subset of the T-REx dataset[2] (Elsahar et al., 2018) created by randomly sampling 1,000 sentences for each relation occurring at least 1,000 times in the original dataset. Finally, a subset of the Semeval 2010

[2] https://hadyelsahar.github.io/t-rex/

Task 8 dataset[3] (Hendrickx et al., 2010) is taken into account, by sampling the 20% of the train set and using it for validation. This dataset is referred to as **SemEval 2010 Task 8 (custom)**.

Standard Evaluation

The standard evaluation for RE takes places on the test split of the generated dataset. Formally, the standard accuracy of an RE model f taking in sentence s in natural language can be defined as

$$Acc(f) = \frac{1}{|D_{test}|} \sum_{(s,r) \in D_{test}} v((s,r), f)$$

where v is the F1 score between the correct relation r and the relation predicted by $f(s)$. In order to allow for a more detailed comparison, the $Recall_{macro}$ and $Precision_{macro}$ are computed.

Adversarial Framework

To compute the accuracy of models under pressure, *adversarial accuracy* is implemented, as defined for NLP purposes by Jia et al. (2017). This accuracy introduces an adversary function A in the computation. A takes in an example (s,r), and returns a new example (s',r') by adding noise to training examples. Adversarials have been proven useful in evaluating models for computer vision (Goodfellow et al., 2015; Szegedy et al., 2014).
The new accuracy is then calculated as

$$Adv(f) = \frac{1}{|D_{test}|} \sum_{(s,r) \in D_{test}} v(A(s,r), f)$$

As highlighted by Jia et al. (2017), for this accuracy to be meaningful, it is necessary for the adversary function to satisfy two requirements:

1. it should generate *valid* tuples (s',r') (i.e., a human should judge r' as being the correct relation given s');

[3] https://github.com/sanitya0000/Relation-Classification

2. (s', r') should be somehow close to the original tuple (s, r).

In this work, the adversarial examples generated are *semantically close* to the original sentences: formally, the objective is, given an initial example (s, r), to generate adversarials $s' \neq s$, while keeping $r' = r$.

The theoretical assumption behind the adversarials generated for this experiment is that, while information about domain and range of a relation are useful in determining the *truth value* of a sentence, they should not play such a critical role in predicting the relation label realized by the sentence.

For instance, given the example in Fig. 12, it is clear that, while adversarial examples describe false or impossible situations, all the sentences express the same relation between `subj` and `obj`.

Original sentence:

$\boxed{\text{Leonardo da Vinci}_{\texttt{sub}}}$ painted the (Mona Lisa)$_{\texttt{obj}}$

Adversarials:

(sub1) $\boxed{\text{Michelangelo}_{\texttt{subj}}}$ painted the (Mona Lisa)$_{\texttt{obj}}$

(sub2) $\boxed{\text{Barack Obama}_{\texttt{subj}}}$ painted the (Mona Lisa)$_{\texttt{obj}}$

(sub3) $\boxed{\text{Stratolaunch}_{\texttt{subj}}}$ painted the (Mona Lisa)$_{\texttt{obj}}$

(sub4) $\boxed{\text{[MASK]}_{\texttt{subj}}}$ painted the (Mona Lisa)$_{\texttt{obj}}$

Figure 12: Examples of generated adversarials

Similar situations can arise when the model is tasked to predict the relation in a sentence with unknown or unexpected tokens. This is the case, for instance, of neologism, in which case the model's overreliance on specific semantic features could be an impairment rather than a benefit. Similarly, not all entities might be encoded correctly by the model, as it might be the case, for instance, of speech-to-text outputs.

In such cases, the model should rely on syntactic information and tokens behind entity mentions in order to predict the correct label, which can be implemented in the form of graph-based knowledge integration.

Based on this assumptions, different substitution strategies are implemented to create a new adversarial dataset by using entity types and domain/range information that can test models' reactions to specific permutations.

First, for every entity mention e appearing in the dataset their corresponding entity e^{WIKI} stored in FewRel is retrieved, together with their entity types e^T, extracted by querying the Wikidata sparql endpoint[4].

To generate the adversarial examples, every sentence-relation pair in the testing split $(d_k, r_k) \in D_{test}$ is taken into account. As previously mentioned, the sentence d_k has two entity mentions $e_{k,i}$, each with a semantic role $e^{SR}_{k,i} \in (subj, obj)$ and a type $e^{T}_{k,i}$.

For each of these pairs in the testing split, a list of possible candidate entities E is generated, which will be used to substitute $e_{k,i}$. This list is retrieved by looking at all the sentence-relation pairs in the training split $(d_j, r_j) \in D_{train}$. As it was the case for d_k, each d_j has two entity mentions $e_{j,i}$, each with a semantic role $e^{SR}_{j,i} \in (subj, obj)$ and a type $e^{T}_{j,i}$.

While it would be possible to use the full list of entities thus retrieved to randomly substitute the original $e_{k,i}$, the objective of this experiment is to investigate specific entity-related phenomena. In order to do so, the list of candidate new entities is filtered by applying one of four strategies, in turn.

The strategies define features for the new entities $e_{j,i}$ and pairs (d_j, r_j) they appears in, and are as follows:

- **sub1**: $e^{SR}_{j,i} = e^{SR}_{k,i} \land r_j = r_k$: the new entity has the same semantic role as the original one, and it appears in sentences expressing the same relation of the original sentence;
- **sub2**: $e^{T}_{j,i} = e^{T}_{k,i} \land r_j \neq r_k$: the new entity is of same type as the original entity, but it never appears in a sentence expressing the relation expressed by the original sentence;
- **sub3**: $e^{T}_{j,i} \neq e^{T}_{k,i} \land r_j \neq r_k$: the new entity is of different type than the original entity, and it never appears in a sentence expressing the relation expressed by the original sentence.

Once the list of candidate entities is generated, one is randomly selected and used to replace $e_{k,m}$. In addition to this, a fourth strategy is defined, where:

- **sub4**: $e_{new} = [MASK]$: the new entity mention is represented by the `[MASK]` token.

This process is repeated three times for each strategy: once for the `subj`, once for the `obj` and once for both entity mentions, thus resulting in 12 newly generated sentences for each of the original testing examples. Table 8 shows an example of all these strategies in action.

[4] https://query.wikidata.org/sparql

original sentence	
relation	**sentence**
P26 - spouse	Derek Denton has been married since 1953 to Dame Margaret Scott
sub1	
entity	**example**
subj	Sigismund I the Old has been married since 1953 to Dame Margaret Scott
obj	Derek Denton has been married since 1953 to Marie of Romania
subj+obj	Gore Vidal has been married since 1953 to Jesse Owens
sub2	
entity	**example**
subj	Åse Gruda Skard has been married since 1953 to Dame Margaret Scott
obj	Derek Denton has been married since 1953 to Lynda Day George
subj+obj	His wife has been married since 1953 to Phila
sub3	
entity	**example**
subj	Canadian yearly meeting has been married since 1953 to Dame Margaret Scott
obj	Derek Denton has been married since 1953 to New Christian
subj+obj	July has been married since 1953 to NGC 362
sub4	
entity	**example**
subj	[MASK] has been married since 1953 to Dame Margaret Scott
obj	Derek Denton has been married since 1953 to [MASK]
subj+obj	[MASK] has been married since 1953 to [MASK]

Table 8: Example of entries from the dataset

4.1.2 Experiment

Text-based models

As for many other NLP tasks, the use of neural representations trained on purely textual data is the default option for RE. In this experiment, the robustness of such text-based models is investigated, in particular by taking into account fine-tuned systems, using either static or contextual embeddings. For static embeddings, the 100-dimensional word embeddings proposed in Turian et al. (2010) are implemented, while for contextual embeddings the 768-dimensional, 110M-parameters `bert-base-uncased` for English language is employed (Devlin et al., 2019). More details about these models are found in Section 4.2.

Graph-based models

Given the objective of investigating whether structured representation of texts is useful in standard RE setting and in stressful situations, graph-based representations are implemented. Furthermore, this sort of representation also eases the introduction of additional knowledge on top of textual representation, as previously described in Section 2.4. These graphs are encoded using knowledge graph embedding models, and the type of information stored is investigated according to the following dimensions:

1. the initial word embeddings implemented for the token nodes;
2. the type of KGE implemented;
3. the type of linguistic information encoded by the graph-based representation;
4. the type of external information included on top of the base graph-based linguistic representation;
5. the structure used to include external information.

These dimensions are first evaluated on their own, and then their combinations are evaluated as well. More information on these models are found in Section 4.2.

4.2 Neural Models for Relation Extraction

This section describes the neural models for RE investigated in the course of this experiment. In particular, they are classified according to the type of representation implemented, which can be either based on just texts, or based on graphs.

4.2.1 Text-based Relation Extraction

These models take a text as an input and then, based on features extracted from co-occurrences, predict a relation label between two entities.

Representation

In order to ease the implementation of the chosen models, the original data from FewRel is first mapped to a more task-specific textual representation, namely the one employed in the course of the Semeval-2010 task 8 for multi-way classification of semantic relations (Hendrickx et al., 2010). Each sentence is thus represented as follows:

(7) 47600 "It's located in <e1>Tondi </e1>, a subdistrict of <e2>Kristiine </e2>.
P276
Comment:

With each example having the following information:

- An id corresponding to the specific example;
- a sentence with <e1>and <e2 >tags annotating, respectively, the subject and the object of the relation;
- the relation label to be predicted (according to the Wikidata ontology[5]);
- a comment section (which is never used for the present work).

The training, test and validation splits of the dataset are all converted into such a format and used to train and evaluate the text-based RE models.

[5] https://www.wikidata.org/wiki/Wikidata:WikiProject$_{\text{O}}$ntology/Modelling

Models

Several text-based neural models for RE are selected as representative of this kind of system.
The total fine-tuning of all the models took around three days on an average GPU. Unless stated otherwise, all the hyperparameters are set according to the original papers.

The first text-based model is represented by an Attention-based biLSTM model (**ATT-biLSTM**), based on the work by P. Zhou et al. (2016a). The model is made up of 4 layers: first, an embedding layer maps the input text into vector space, then two LSTM layers are implemented on top of it, and finally the output of the two LSTM layers are put through an attention layer. This model is trained end-to-end for 50 epochs, starting from word vectors proposed in Turian et al. (2010))[6].

The second text-based baseline is a convolutional neural network (**CNN**) trained on RE, inspired by D. Zeng et al. (2014). As per the original paper, the first step of the model is to represent the word as vectors using pretrained embeddings by Turian et al. (2010).
These word vectors, together with position features expressing the distance of the token from the two entity mentions, are fed into a convolutional component to extract sentence-level features, which are then combined with lexical-level features about the entity mentions (i.e., their vector representations, together with their adjacent tokens' representations and vector space representation of their hypernyms according to Wordnet) to generate the final representations used to predict the relation label.
The model is trained for 20 epochs, with 0.5 dropout and 1e-3 learning rate.

The next set of models is composed of different methodologies for fine-tuning transformer models. For all these models, word embeddings are initialized using the BERT base uncased model.
The first one of such models is BERT with entity markers (**BERT$_{\mathbf{em}}$**), one of the models proposed by Baldini Soares et al. (2019)[7]. This model adds entity tags before and after each entity, and then it represents the relation between the entities as the concatenation of the start tag for each of the entity mention. This concatenation is used as input to a fully connected layer, which predicts a relation label. This model is fine-tuned for 5 epochs, with 3e-5 learning rate.

[6] https://www.kaggle.com/datasets/alvations/turian-embeddings

[7] This model is implemented using the unofficial implementation found at https://github.com/plkmo/BERT-Relation-Extraction

The second one of such models is **R-BERT** (S. Wu et al., 2019)[8] with the same parameters found in the original paper. This model makes use of mentions-specific embeddings by averaging over the embeddings of the tokens composing each mention and using them as input to a fully connected layer. Then, it concatenates the final outputs of each mention to the representation of the `[CLS]` token. This is then used as an input for a softmax layer to predict relation labels. This model is fine-tuned for 5 epochs with 2e-5 learning rate.

The final model is represented by **RIFRE**, as proposed by Zhao et al. (2021)[9]. This model uses both pretrained word embeddings for each token, and relation embeddings for each relation, which are used in an heterogeneous graph representation composed of word nodes and relations nodes. In this graph, nodes of one type are all considered adjacent of nodes of the other type, and these are the only edges present in the graph. Message passing is then applied on top of this graph in order to update word embeddings and use them to classify the expressed relation, thus iteratively fusing information about word and relations.
While this model makes use of an implicit graph representation, its edges are still related to information related to the distribution of tokens (i.e., which words appear with each relation with a higher or lower frequency). As such, this model is included among text-based representations.
This model is fine-tuned for 5 epochs and 1e-1 learning rate.

4.2.2 Graph-based Relation Extraction

These models are defined as models that take explicit graph-based representations of texts as inputs, and predict a relation label.

Graph-based Representations

The base structure for the graph-based representation implemented in this work encodes tokens as nodes, and then builds relations between them, according to what kind of information is encoded. In order to include information regarding word meaning and their use in context, each node is initialized using word embeddings. More specifically, the use of either **BERT**_base_uncased, or **Glove** is

[8]This model is implemented using the unofficial implementation found at https://github.com/monologg/R-BERT

[9]This model is implemented through the official representation found at https://github.com/zhao9797/RIFRE

explored as possible initial embeddings for the tokens' nodes.
This work investigates several possible graph structures, described in the following paragraphs.

Chain-graph In this setting, the only relation encoded is whether two tokens occur near one another. For instance, given the sentence *Leonardo da vinci painted the Mona Lisa*, it would be represented as shown in Figure 13.
This kind of graph explicitly represents information about word co-occurrences,

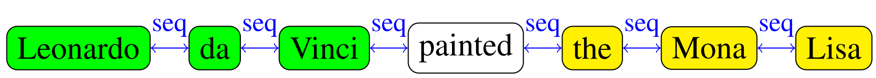


Figure 13: Example of a chain-graph. Entity mentions are colored according to the entity they refer to.

and as such about the context of tokens themselves.

Syn-graph This representation is based on information extracted from a syntactic dependency tree, easily retrievable by means of an off-the-shelf NLP tool[10]. This structure represents the most common method found in the literature for GCN.
An example of **syn-graph** is shown in Figure 14.

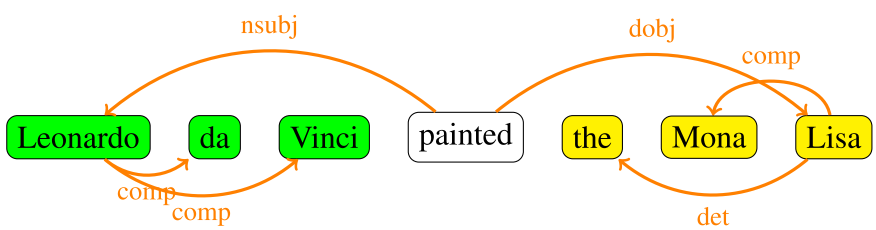


Figure 14: Example of a syn-graph. Entity mentions are colored according to the entity they refer to. The labels are as follows: nsubj stands for nominal subject, comp stands for compound, dobj stands for directobject, and det stands for determiner.

[10]For this experiment Spacy library for Python is used.

Sem-Graph Inspired by the work by Marcheggiani et al. (2018), the integration of semantic information is tested, in the form of PropBank-style (Palmer et al., 2005b) semantic role structures. By extracting the arguments for the sentences' predicates, it is possible to build a dependency-based graph for semantic roles (Hajič et al., 2009) where each predicate is connected to its semantic arguments, with each edge thus created labeled with the encoded role. An example of **sem-graph** is shown in Figure 15.

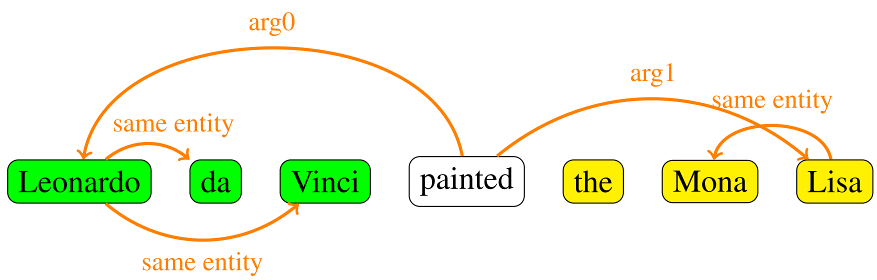


Figure 15: Example of a sem-graph. Entity mentions are colored according to the entity they refer to. The labels are as follows: arg0 describes the predicate's agent and arg1 the patient.

Combinations The relations previously described can be re-combined to create more complex graphs. The following combinations are investigated:

1. chain graph;
2. syn graph;
3. sem graph;
4. chain+syn graph;
5. chain+sem graph;
6. syn+sem graph;
7. chain+syn+sem graph.

An example of a `chain+syn+sem graph` is shown in Figure 16.

Hyperparameters The implementation of several hyperparameters for graph representation is also investigated, by following proposals present in the literature. In particular, the followings parameters are investigated:

- **Reverse edges** Reverse edges can be added, so that for any relation r there also exists an opposite-direction relation r', as described by Marcheggiani et al. (2017)

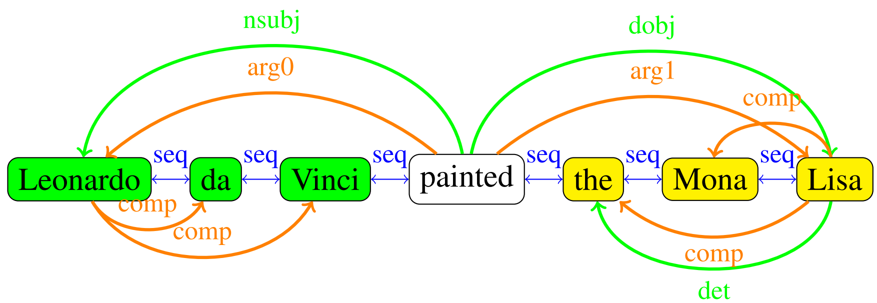


Figure 16: Example of a chain+syn+sem graph. Entity mentions are colored according to the entity they refer to.

- **Self-loops** Self-loops, namely relations starting from a node and ending at the same node, can be added so that information about a node (i.e., its initialized embeddings) is included in the computation of the final representation, as observed by Kipf et al. (2016).
- **Edge labels** While structural information is encoded by edges between nodes, these edges can be labeled in order to have a more fine-grained representation of edges in the graph.

Graph-based integration of external information

The integration of non-linguistic information in graph-based representations for RE is also investigated. In particular, the experiments are carried out with regards to the integration of information from a background Knowledge Graph (in particular, Wikidata[11]), and the integration of information from a graph-based lexicon (in particular, Wordnet[12]).
More specifically, Wikidata can be used to retrieve information about entities involved in the relation realized by a specific sentence, while WordNet can be used to leverage linguistic information about tokens by extracting the corresponding synsets.
Since both the resources are represented as graph structures, KGE algorithms are first implemented in order to extract vector-space representations for the entities present in the resources. In particular, these distributional representations are computed using the RDF2Vec algorithm (Ristoski et al., 2018) and then included to-

[11]https://www.wikidata.org/wiki/Wikidata:Main\page
[12]http://wordnet-rdf.princeton.edu/

gether with the initial nodes' embeddings.
These features are added as additional nodes connected to their associated token/entity, and they share the same dimensionality as the initial node/token features, so that they can be easily used in the GNNs.

Knowledge Graph Embeddings

Over the years several algorithms have been proposed in order to represent graphs' structural information in a vector space, to then use the latter in a downstream task. As such, different such algorithms are tested in the present work.
Once the nodes are represented in such a format, they are fed to a Feed Forward Neural Network which then classifies relation labels through a logistic regression classifier, as defined in Guo et al. (2019).
Several Graph Neural Networks are evaluated, namely Relational Graph Attention Networks, Graph Convolutional Networks and Relational Graph Convolutional Networks, with the main hyperparameters being the dimensionality of the layers (64, 120, 240), the learning rate (8 samples from 0 to 10^{-10}, and the batch size (32,64,128).

Relational Graph Attention Networks Relational Graph Attention Network (RGAT) is an extension of graph attention mechanism used to incorporate relational information (Busbridge et al., 2019). This model is built based on the attention mechanism described in Vaswani et al. (2017) and the GAT layer described in Veličković et al. (2017). The core idea of RGAT is that, while GAT aggregates the representations of neighborhood nodes along paths without taking into account which relation is encoded by each path, different relations should have different influence on the final representation (K. Wang et al., 2020). This is done through the integration of an additional attention head, which encodes attention towards specific relations. This head is concatenated to the original GAT attention head, and then integrated in the final representation through a softmax function.

Graph Convolutional Network for NLP Graph Convolutional Network (GCN) as proposed by Kipf et al. (2016) represents a model used to incorporate information about adjacent nodes in nodes' representations (see Section 2.4). In this experiment, the generalization proposed by Marcheggiani et al. (2017), Marcheggiani et al. (2018) for NLP tasks is implemented. In this implementation, in order to incorporate directions and labels, the authors propose the integration of direction-specific weights and relation-specific biases in the original GCN formulation

Relational GCN The final model is a Relational Graph Convolutional Network (RGCN), a widely adopted model for KGE (Thanapalasingam et al., 2022) proposed by Schlichtkrull et al. (2018). This model is an extension of the basic GCN, implemented in order to include relational graphs with labeled edges.

4.3 General Results

This section investigates the results of the RE models against the original dataset. First, the models based on text-based representations are taken into account, then the models based on graph-based representations.

4.3.1 Text-based models

Text-based models perform extremely well on the RE task, with even the most basic model (Att-BiLSTM) reaching an F1 score higher than 0.60.
The models based on BERT fine-tuning reach the highest scores, and the best performing one is RIFRE, showing that an implicit graph representation has a positive effect on the RE task.

Model	**Acc**
Att-BiLSTM	.66
CNN	.71
BERT_{em}	.84
R-BERT	.89
RIFRE	**.90**

Table 9: General evaluation of text-based models

4.3.2 Graph-based models

Given the previously defined hyperparameters of graph-based models, an evaluation is carried out in order to find out the best-performing model. In particular, the combinations are evaluated using ASHA (L. Li et al., 2020), a framework for hyperparameter search that applies intelligent early-stopping, while supporting large-scale parallelization.
The best-performing model architecture across all experiments is based on a two-layer R-GCN encoder with self-loops and reverse edges, with hidden dimension of 120 for Glove and 240 for BERT initial embeddings. Moreover, the best learning rate is usually around $0.3e^{-3}$, and batch size does not have an impact on the final model and can be chosen based on the given computing resources.

As shown in Table 10, basic graph-based representations using Glove embeddings do not show good results for RE. In particular, the best Glove-based graph model reaches 0.561 F1, 10% worse than the lowest-scoring text-based model.
Models using BERT features, on the other hand, perform generally better on the standard test set, in all cases of graph configuration.

Model	**Glove**				**BERT**			
	acc	f1	p	r	acc	f1	p	r
chain (chaingraph)	.52	.5	.508	.521	.7	.688	.689	.7
syn. (syngraph)	.589	.575	.573	.589	**.740**	**.739**	**.745**	**.740**
sem. (semgraph)	.457	.441	.443	.457	.616	.59	.585	.616
chain + syn.	.580	.565	.561	.58	.727	.723	.725	.727
chain + sem.	.565	.552	.556	.565	.705	.699	.701	.705
syn. + sem. (synsemgraph)	**.604**	**.589**	**.588**	**.604**	.744	.740	.741	.744
chain + syn. + sem.	.58	.56	.56	.58	.728	.725	.729	.728

Table 10: General evaluation of graph configurations. *chain* denotes the next-token relation, *syn.* denotes syntactic dependencies, and *sem.* denotes semantic dependencies.

Overall, for the models trained on just one graph type (without external knowledge), the best performance is reached with syntactic dependencies using BERT features, with an F1 score of 0.739, followed by models trained with a linear chain of tokens (F1 of 0.7 using BERT tokens), followed by models trained with semantic dependencies (F1 of 0.616 using BERT features). Even though the chain graph on its own leads to acceptable results, its addition to graph structures leads to worse results.
Our hypothesis is that the linear chain shifts the attention away from relevant de-

pendency information (either syntactic or semantic) to the linear chain information, which actually adds no additional knowledge. However, the combination of syntactic and semantic dependencies shows the best results for combining graph structures, achieving an F1 score of 0.74. An evaluation of this model is further carried on using other datasets, as shown in Table 11.

Model	**Glove**				**BERT**			
	acc	f1	p	r	acc	f1	p	r
FewRel (custom)	.604	.589	.588	.604	.744	.740	.741	.744
T-REx (custom)	.445	.39	.368	.445	.69	.658	.644	.69
SemEval 2010 Task 8	.786	.786	.789	.786	.778	.777	.779	.778

Table 11: Scores for the datasets: FewRel (custom), T-REx (custom), and SemEval 2010 Task 8, based on the synsem graphs.

Moreover, the impact of additional knowledge graph features is evaluated. As a baseline, a neural network with two linear layers and a classification layer is used. This model takes as inputs the concatenated *RDF2Vec* features (*Glove* 2 x 100, *BERT* 2 x 758), and it is trained to predict the correct relation annotated in our FewRel dataset. The results, in relation to the best-performing graph structure *synsem*, are shown in Table 12.
As expected, the integration of additional features from a background KG leads to an improvement of the scores in all cases. In particular, the best GNN models make use of information from Wikidata. For instance, using Wikidata's shortest paths on *Fewrel (custom)* achieves an F1 score of 0.834, which is an increase of 0.094 compared to the base model. Altogether, all GCN models using Wikidata features outperform the baseline neural network using Wikidata RDF2Vec features. Even though WordNet features increase the performance of models, these models only rarely outperform the Wikidata RDF2Vec baseline, proving that world knowledge is more beneficial to these models compared to linguistic knowledge.

Model	Glove			BERT		
	f1	p	r	f1	p	r
	FewRel (custom)					
Linear-NN:w. emb.	.277	.323	.294	.382	.436	.3
Linear-NN:RDF2Vec	.597	.618	.606	.664	.675	.669
GCN:synsemgraph	.589	.588	.604	.740	.741	.744
+Wikidata nodes	**.783**	**.782**	**.799**	.821	.824	.826
+Wikidata s.p.	.405	.381	.459	**.834**	**.835**	**.826**
+WordNet nodes	.602	.604	.612	.726	.729	.703
	T-REx (custom)					
Linear-NN:BERT	.082	.15	.103	.239	.304	.251
Linear-NN:RDF2Vec	.438	.488	.475	.506	.548	.525
GCN:synsemgraph	.39	.368	.445	.658	.644	.69
+Wikidata nodes	**.61**	**.574**	**.678**	**.718**	**.699**	**.76**
+Wikidata s.p.	.425	.407	.475	.671	.659	.7
+WordNet nodes	.419	.402	.465	.654	.65	.685
	Semeval 2010 Task 8					
Linear-NN:BERT	.548	.675	.52	.629	.694	.638
GCN:synsemgraph	**.786**	**.789**	**.786**	**.777**	**.779**	**.778**

Table 12: Evaluation of additional KG features added to *synsemgraph*. The abbreviation **s.p.** denotes the *shortest paths* configuration.

4.4 Adversarial Evaluation

4.4.1 Robustness of models

After choosing the best GCN models based on the standard test set, they are compared to established LSTM, CNN and Transfomers models on the adversarial test set. The results are shown in Table 13, with **acc** describing models' results to standard RE, **adv** showing adversarial evaluation of models, **diff** the difference between the two evaluations. The following columns contain data about F1 score for specific strategies.

Similarly to general results, text-based models fare relatively well to adversarial examples. The best model performance-wise is RIFRE, with an F1 of 0.727, -0.173 compared to the results on the original set.

Generally speaking, all the text-based models lose at least 0.15 points in F1 score in an adversarial environment, with the least robust model being the Att-BiLSTM model with -0.22 on its adversarial F1, and the most robust being the CNN mode and BERT_em both with -0.15.

Against the expectations, the GCN models generalize worse than the text-based models, showing weak robustness against data permutations. In particular, even the best-faring GCN model (*syngraph*) only generalizes slightly better than the worst generalizing Att-BiLSTM model (-0.30% and -0.33%, respectively). This refutes the initial hypothesis that GCN models, and graph-based representations in general, show good generalizability and robustness across standard NLP tasks.

4.4.2 Strategy-wise Analysis

By investigating results on the basis of the implemented strategies, certain patterns are found across all models, showing that different models react in similar ways to certain permutations.
The results, shown in Table 13, take into account every text-based models, while for graph-based representations only the four base graph structures are considered, namely chain graphs (**chaingraph**), syntactic graphs (**syngraph**), semantic graphs (**semgraph**), and the combination of both syntactic and semantic graphs (**synsemgraph**). These configurations are used to train a RGCN starting from BERT embeddings, since they result in the best performance given ASHA.

Across all substitution strategies, it can be noticed that the substitution of the subject entity mention does not have as strong of an impact as the substitution of the object entity mention on F1 scores.
This is particularly evident for sub3 and sub1, with the former having an average F1 of 0.59 when subj is substituted, 0.36 when obj is substituted and 0.22 when both entity mentions are substituted. Regarding sub4, on the other hand, the models reach an average of 0.61 when the subject is masked, 0.427 when obj is masked, and 0.28 when both entity mentions are masked.

Finally, while the first three substitution strategies show a decreasing level of performance, with sub3 showing worse results than sub2, and the latter showing worse results than sub1, it is interesting to notice that sub4 does not show worse results than sub3. Actually, for specific models (e.g., BERT_{em}) masking entity mentions show improvement over random substitution, proving that models rely on entity mentions to predict relations' labels, and that an incorrect encoding of such entity mentions can lead to issues.

model	**acc**	**adv**	**diff**	**sub1**			**sub2**			**sub3**			**sub4**		
				subj	obj	subj + obj	subj	obj	subj + obj	subj	obj	subj + obj	subj	obj	subj + obj
Att-BiLSTM	.66	.44	-.33%	.68	.62	.41	.65	.58	.57	.55	.28	.20	.51	.16	.06
CNN	0.71	.56	-.21%	.72	.69	.50	.71	.67	.66	.58	.38	.26	.64	.49	.41
R-BERT	.89	.7	-.21%	.88	.86	.57	.87	.84	.83	.74	.43	.30	.83	.66	**.56**
RIFRE	**.90**	**.7**	-.2%	**.89**	**.87**	.62	**.88**	**.85**	**.83**	**.77**	**.54**	**.39**	**.85**	**.73**	.51
BERT_em	.84	.69	-.18%	.85	.83	**.64**	.84	.81	.79	.72	**.54**	.37	.75	.63	.53
chaingraph	.69	.42	-.39%	.70	.64	.30	.67	.59	.57	.48	.23	.08	.47	.24	.06
syngraph	.74	.46	-.3%	.73	.67	.34	.71	.63	.58	.54	.30	.13	.49	.30	.08
semgraph	.59	.4	-.32%	.58	.54	.31	.58	.50	.50	.44	.24	.14	.49	.31	.22
synsemgraph	.74	.48	-.35%	.73	.67	.37	.71	.64	.59	.55	.32	.15	.53	.33	.12
avg	.75	.54	-,28%	.75	.71	.45	.73	.67	.65	.59	.36	.22	.61	.43	.28

Table 13: Fine-grained F1 scores for the models. GCN uses *BERT*.

Chapter 5

Question Answering over Structured Data

5.1 Task Definition

As already mentioned, the large amount of data continuously published on the Web directly leads to an increase in the demand for multilingual tools and user interface to simplify the access and reusability of data by end-users. More noticeably, there is a strong interest in the construction of models that enable the querying of data through natural language utterances, as proven by the many QA systems developed over the years (Section 1.1.2).
As with many other NLP tasks, the default approach for QA systems is represented by neural models trained over huge task-specific datasets, such as the Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016; Rajpurkar et al., 2018) for IR models, or WikiSQL for KBQA (Zhong et al., 2017).
While models trained in such a way generally show impressive results[1], they also suffer from major drawbacks:

[1]See, for instance, SQuAD leader board at https://rajpurkar.github.io/SQuAD-explorer/ and WikiSQL leader board at https://github.com/salesforce/WikiSQL

- they need an extremely large datasets to be trained on, making the adaptation of systems to other domains and languages extremely demanding[2];
- neural models show a lack of controllability, for instance it is unclear which new examples are to be added to make a new question answerable;
- they inherit the general opaqueness of neural networks.

These issues are particularly apparent when dealing with QA over structured data (Chakraborty et al., 2019), since natural language questions first have to be mapped through semantic parsing to a formal representation that can be used over a knowledge base, which adds an additional step of complexity to the architecture. Furthermore, the initial effort of providing training data for such semantic parsing in a neural environment is extremely demanding.

This chapter describes two efforts made in relation to the task of QA over graph-based structured data. The first system, described in Section 5.2, is the subject of discussion of Nolano et al. (2021), and represents an extension for Italian language of the model-based approach to QA previously described by Benz et al. (2020) and Elahi, Ell, et al. (2021). This approach is based on the generation of grammars using lemon lexica[3], that can then be employed for the interpretation of questions by parsing them into SPARQL queries according to semantic and syntactic information. This approach has been shown to scale well to large numbers of questions, as the system developed for the Italian language can answer more than 1.6 million Italian questions over the DBpedia knowledge graph.

The second model (see Section 5.3) is represented by *Across Europe with Maggie*, a methodology developed in the course of the European Datathon 2022[4], where it was awarded the third place in Challenge 4: a Europe Fit for Digital Age. The core module of the model is implemented using the Rasa platform[5], which implements entity and intent recognition through a neural architecture. The results are then fed into a semantic parser, and used to answer questions over data from several sources (mainly the Europeana portal and European open data[6]). In order to make use of state-of-the-art technology in relation to graph traversing, the scraped data is mapped to a labeled property graph using the Neo4j platform[7].

[2]See Croce et al. (2018) for a description of the efforts involved in the creation of the Italian version of SQuAD.

[3]https://lemon-model.net/

[4]https://op.europa.eu/en/web/eudatathon

[5]https://rasa.com/

[6]https://data.europa.eu/en

[7]https://neo4j.com/

5.2 An Italian QA system based on grammars

The core module for the QA system described in Nolano et al. (2021) is represented by the model-based system proposed in Benz et al. (2020) and extended in Elahi, Ell, et al. (2021).
This system uses Linked Open Data as a source of knowledge, and it implements a lemon lexica-based representation (J. McCrae et al., 2011) to specify how the vocabulary elements of an ontology or a knowledge graph (e.g., entities and relations) are realized in natural language questions.
The first extension of such a module for the Italian language has been proposed in the course of the *Hackathon on Question Answering based on automatically generated grammars*[8] (5-9 July 2021), where an extension is implemented in order for the model to answer CH-related questions in Italian language (Elahi, Nolano, et al., 2021) over a domain-specific knowledge graph.
A later published paper (Nolano et al., 2021) describes a more general-domain extension for the Italian language, in order to cover natural language questions over DBpedia entries. These efforts are described in the following sections.

5.2.1 Methodology

The core structure of this model is represented by a grammar generator module, which is then adapted to Italian language by integrating language-specific questions and patterns.

The grammar generator

The grammar generation module is based on a mapping between syntactic constructions used in questions and data from a given ontology and/or knowledge graph in the form of classes and properties. More specifically, the process makes use of several *frames*, each describing the linguistic realizations of specific properties which might appear in a question.
In particular, the following frames are implemented:

- NounPPFrame
- TransitiveFrame

[8] https://scdemo.techfak.uni-bielefeld.de/qahackathon/index.php

- IntranstivePPFrame
- AdjectiveAttributive
- AdjectiveGradable

For instance, the DBpedia property `dbo:capital`, which has `Country` as its domain and `City` as its range, refers to the semantic relation existing between an entity referring to a country and an entity referring to its capital, and is realized by the NounPPFrame *'Y is the capital of X'* in natural language.
This frame then leads to the generation of the following questions:

- What is the **capital** of X(`Country`)?
- Which city is the **capital** of X(`Country`)?

Such a rule can be used to both generate natural language questions that could be used to refer to specific properties, while also posing semantic constraints on the kind of entities and labels that can appear in questions.
Similar rules can also be defined for transitive constructions (TransitiveFrame), intransitive verbs (IntranstivePPFrame), as well as adjective constructions in both attribute (AdjectiveAttributive) and predicate (AdjectiveGradable) forms.
In the context of the work defined in Nolano et al. (2021), the generation of rules has been adapted to the Italian language, by building on these frames and making them usable in an Italian environment.

Italian Adaptation

As an example for the actual adaptation of the model, the Italian lemon entry for the property `dbo:capital` is shown in Figure 17. This entry states that the canonical written form for the property is *capitale* and that the entry has a NounPPFrame as its syntactic behavior. It also states that it takes two arguments, the first being the object of the property, and the second its subject, and that the natural language representation for this property is realized through the *di* (of) preposition. This last information is useful for the answer generation part of the model. In order to adapt the original model to Italian, language-specific phenomena had to be accounted for for question realizations. In particular, the following properties had to be accommodated for:

- sentence order;
- auxiliary verbs;
- interrogative pronouns and adjectives;
- the use of different prepositions on the basis of range/domain semantics;
- the use of determiners on the basis of range/domain semantics.

```
:lexicon_en a    lemon:Lexicon ;
  lemon:language "it" ;
  lemon:entry    :capital_of ;
  lemon:entry    :di .

:capital_of a          lemon:LexicalEntry ;
  lexinfo:partOfSpeech lexinfo:noun ;
  lemon:canonicalForm  :capital_form ;
  lemon:synBehavior    :capital_of_nounpp ;
  lemon:sense          :capital_sense1 .

:capital_form a    lemon:Form ;
  lemon:writtenRep "capitale"@it .

:capital_of_nounpp a           lexinfo:NounPPFrame ;
  lexinfo:copulativeArg        :arg1 ;
  lexinfo:prepositionalAdjunct :arg2 .

:capital_sense1 a lemon:OntoMap, lemon:LexicalSense ;
  lemon:ontoMapping      :capital_sense1 ;
  lemon:reference        dbo:capital ;
  lemon:subjOfProp       :arg2 ;
  lemon:objOfProp        :arg1 ;
  lemon:condition        :capital_condition .

:capital_condition a   lemon:condition ;
  lemon:propertyDomain dbo:Country ;
  lemon:propertyRange  dbo:City .

:arg2 lemon:marker :di .

:di a                  lemon:SynRoleMarker ;
  lemon:canonicalForm  [ lemon:writtenRep "della"@it ] ;
  lexinfo:partOfSpeech lexinfo:preposition .
```

Figure 17: Lemon entry for the relational noun *'capitale della'*

The syntactic patterns developed for Italian, together with question samples, are shown in Table 14. More specifically, they are implemented using the tagset from the Penn Treebank Project (Marcus et al., 1993), with V* defining all possible forms of a given verb, words in brackets defining nounsverbsadjectives that realize a specific property, and `dbo:range/dbo:domain` defining specific labels representing classes, e.g., `dbo:Country` can be represented by either *paese* (country) or *nazione* (nation).
In the next sections the behavior of different syntactic frames for the Italian module is illustrated.

NounPPFrame Assuming that in the corresponding lemon lexicon the connection between the NounPP construction *capitale della* (capital of) is modeled as referring to the property `dbo:capital` with domain `country` and range `city`, the following questions can be generated automatically:

1. *Qual è la capitale della* (what is the capital of) (X|country_NP)?
2. *Quale città è la capitale della* (which city is the capital of) (X|country_NP)?

where X is a placeholder that can be substituted by any particular country, e.g.

LexInfo Frame	Syntactic Pattern	Question Sample
NounPP	WDT/WP V* DT [noun] IN DT [domain]	*Qual è la **capitale della** Germania?*
	WDT `dbo:range` V* DT [noun] IN [domain]?	*Quale città è la **capitale della** Germania?*
	WDT/WP V* DT [noun] in [domain]	*Chi era la **moglie di** Abraham Lincoln?*
	[range] V* DT [noun] IN (DT) [domain]	*Rita Wilson è la **moglie di** Tom Hanks?*
AdjectiveAttributive	WDT V* DT `dbo:range` [adjective]	*Chi era un vescovo cristiano **spagnolo**?*
	[domain] VB (DT) [adjective]	*Barack Obama è un **democratico**?*
AdjectiveGradable	WRB V* [adjective] DT [domain]	*Quanto è **lungo il Barguzin?***
	WDT V* DT [domain] JJS IN (DT) [range]	*Qual è la montagna **più alta** della Germania?*
Transitive	WP V* [domain]	*Chi **ha scritto** Ziggy Stardust?*
	WDT `dbo:range` V* [domain]	*Quale cantante **ha scritto** Ziggy Stardust?*
	WP V* DT [domain]	*Chi **ha fondato** C&A?*
	WDT `dbo:range` V* DT [domain]	*Quale persona **ha fondato** C&A?*
	[domain] V* [range]	*Socrate **ha influenzato** Aristotele?*
IntransitivePP	WRB VB [domain]	*Quando **è iniziata** l'operazione Overlord?*
	IN WDT `dbo:domain` VB [range]	*In quale data **è iniziata** l'operazione Overlord?*
	WDT `dbo:domain` VB IN [range]	*Quale libro **è stato pubblicato** nel 1563?*
	[domain] V* IN [range]	*Il libro dei martiri di Foxe **è stato pubblicato** nel 1563?*

Table 14: Italian Patterns and Questions

Germania (Germany), or by any noun phrase describing a country, such as *paese dove si parla tedesco* (the country where German is spoken). Furthermore, the underlined token can be substituted by entity-specific labels, for instance *città* (city) for an entity of type `city`, or *fuso orario* (time zone) for an entity of type `time zone`.

TransitiveFrame Assuming that the lemon lexicon describes the meaning of the construction *X ha scritto Y* (X wrote Y) as referring to the property `dbp:author` with `song` as domain and `person` as range, the following questions can be generated:

1. *Chi ha scritto* (who wrote) (X|song_NP)?
2. *Quale cantante ha scritto* (which singer wrote) (X|song_NP)?
3. *Quale* (which) (X|song_NP) *è stata scritta da* (was written by) (Y|person_NP)?

Just as X represents a placeholder that can be substituted by entities in the range of the relation, Y defines a placeholder for entities in its domain, and can be substituted by either specific entities, such as *John Lennon*, or by noun phrases describing a particular entity, for instance *l'autore di My Way* (the author of My Way), or *il cantante* (the singer).

IntransitivePPFrame Assuming a structure such as *X pubblicato nel Y* (X published in Y) as a valid representation for the property `dbp:published`, with `song` as its domain and `date` as its range, the following questions can be generated:

1. *Quando è stata pubblicata* (X|song_NP)? (when was (X|song_NP) published?)
2. *Quale* (X|song_NP) *è stata pubblicata nel* (Y|date)? (which (X|song_NP) was published in (Y|date)?
3. *In quale data è stata pubblicata* (X|song_NP)? (in which date was (X|song_NP) published?)

AdjectiveAttributive and AdjectiveGradable Adjectives are divided into two separate frames, according to the type of question generated: AdjectiveAttributive and AdjectiveGradable. The former kind of frame is used to identify that accepts either a boolean answer, or a list of entities; the latter, on the other hand, accepts either a numerical value or a single entity as its answer.

For instance, assuming that the meaning of the (attributive) adjective *democratico* (democratic) is represented by the structures *X è un presidente democratico* (X is a democratic president), the following questions can be generated:

1. *Chi era un presidente democratico?* (who was a democratic president?)
2. (X|person_NP) *è un democratico?* (is (X|person_NP) a democratic?)

While the adjective *democratico* (democratic) can be substituted by any other adjective adhering to this structure, this frame is mainly used for nationality.
On the other hand, assuming that the lemon lexicon would capture the meaning of the (gradable) adjective *lungo* (long) as referring to the property `dbp:length`, the grammar generation module generates the following questions:

1. *Quanto è lungo il* (how long is the) (X|river_NP)?
2. *Qual è il fiume più lungo {del mondo, del Kentucky}?* (what is the longest river in {the world, Kentucky}?)

5.2.2 Results

In Nolano et al. (2021) the proposed methodology is applied to generate natural language questions in Italian language in order to answer questions over the DBpedia dataset. A lemon lexicon comprising 249 lexical entries is created, with the number of grammar rules and questions generated for each frame shown in Table 15.

Altogether, the approach generates 620 grammar rules and covers about 1.6 million questions.

Frame type	**#Entries**	**#Grammar rules**	**#Questions**
NounPPFrame	113	226	1,010,234
TransitiveFrame	41	124	595,854
IntransitivePPFrame	58	116	52,040
AdjectiveAttributiveFrame	29	130	10,025
AdjectiveGradable	8	24	3,123
Total	**249**	**620**	**1,671,276**

Table 15: Frequencies of entries with a certain frame type. The entries are created manually; the rules and questions are generated automatically.

The Italian module was then evaluated using the training split of QALD-7[9]. This dataset contains a total of 214 multilingual questions over linked data, which are matched against the ones generated by the grammar by implementing Jaccard similarity, and then mapped to the property that originally generated the matched question. The initial results (Table 16) show a satisfying precision, but a low recall.

The main reason behind such results is related to the presence of different types of questions in QALD: besides simple questions related to just one relation, QALD also presents complex questions referring to multiple triples, e.g. *A quale movimento artistico apparteneva il pittore de I tre ballerini?* (what was the artistic

[9]https://github.com/ag-sc/QALD

movement of the author of The Three Dancers?), which are not originally covered by the model.
Further work on the Italian module aims at improving these initial results, through

Precision	0.485
Recall	0.224
F-Measure	**0.307**

Table 16: Evaluation results against QALD-7

the integration of new manually-defined frames for specific questions and of a module for the recognition of complex entities. These efforts are the subject of future works on this topic.

5.3 Across Europe with Maggie

Across Europe with MAGGIE is a project proposed in the course of the 2022 edition of the EU Datathon[10], the annual European open data competition organized by the Publication Office of the European Union.
This project aims at developing MAGGIE, a virtual assistant to ease the access to information about European Cultural Heritage. MAGGIE makes use of NLP techniques in order to process input questions, and can aggregate data and make it easy to access through information visualization customizable according to users' interests and geolocated data.

The general workflow for MAGGIE is shown in Figure 18. The two main components for MAGGIE are the background knowledge graph and the QA component used to access it through natural language questions. The next sections describe the assistant in detail.

5.3.1 MAGGIE, a virtual assistant for CH

MAGGIE is a virtual assistant in the form of a real-time chatbot which allows users to explore in an efficient way information about European CH. While the

[10] https://op.europa.eu/en/web/eudatathon

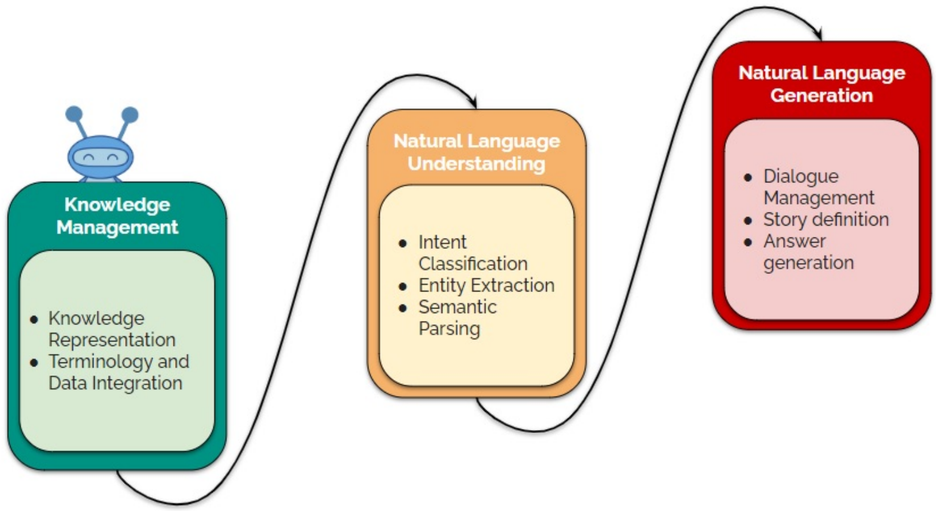


Figure 18: Workflow

majority of the chatbots in the CH domain offer tailor-made experiences for specific museums, one of the main objectives of MAGGIE is to offer its users an experience that covers the entire European territory.
More specifically, MAGGIE provides different access typologies, according to users' preferences: (i) a standard access, allowing users to search for information on the basis of keywords (ii) a registered access, that provides personalized information, tailored to users' preferences set in their profile, (iii) a geolocated access, that makes use of information according to users' location.
Information is fed to the chatbot based on one of these three types of access, allowing for custom-made dialog interaction so that contents can be provided according to users' specific requests, preferences or location. Figure 19 shows a representation of the three different access methods.

5.3.2 Data Collection

The main source of data for MAGGIE is represented by European Open data, in particular by the Europeana Data Collection[11]. The data was accessed and scraped using the Europeana SPARQL endpoint[12]. In order to generate a knowledge base

[11] https://pro.europeana.eu/page/linked-open-data
[12] https://pro.europeana.eu/page/sparql

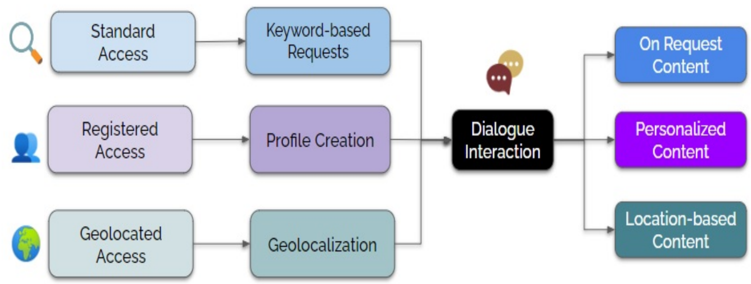


Figure 19: Access methods

to test the methodology on, the results were filtered so that only objects appearing in either Paris or Brussels would be extracted.

Knowledge Representation

While the use of such LOD has been proven useful in many tasks (see Section 1.4.1), in order to make use of optimized algorithms and tools implemented in industry systems, the original data is first mapped to a property graph format. More specifically, this is done using Neo4J[13] through its Python driver[14].
Once the data was scraped from the Europeana Data Collection, the following properties are taken into account for each of the entries[15]:

- `title`: the title of the object
- `dc:Description`: a multilingual description of the object
- `edm:isShownAt`: the uri where the object is stored
- `dc:Creator`: the name of the creator of the object
- `edm:Preview`: a picture showing the object
- `edm:Concept`: an uri describing a concept representing the object
- `edm:ConceptPrefLabelLangAware`: multilingual labels for said concept
- `country`: the country in which the object can be found
- `edm:PlaceAltLabelLangAware`: multilingual labels for the place where the object can be found

[13] https://neo4j.com/

[14] https://neo4j.com/developer/python/

[15] The prefix `dc` refers to the namespace http://purl.org/dc/elements/1.1/, and the prefix `edm` to the namespace http://www.europeana.eu/schemas/edm/.

- `edm:PlaceLatitude` and `edm:PlaceLongitude`: the latitude and longitude of the place where the object can be found
- `europeanaCollectionName`: the name of the collection in which the object is featured as per Europeana's classification

As mentioned already, the main difference between Open Data (i.e., RDF) graphs and Property Graphs is that Open Data graphs do not accept attributes for nodes/relations, so that each of the information listed above is represented as a triple between the original entry's uri and either a specific (possibly empty) node's uri, or a literal.

Labeled property graphs, on the other hand, allow for the definition of nodes' and relations' property, which can improve performance by avoiding the traversal of unconnected data (this is mainly the case for literals in RDF).

Because of this difference, the first step in going from Open Data to PG was to define which information should be kept as nodes and relations, and which should instead be stored as properties.

In the final representation, the original information as described in the Open Data format was re-organized according to the nodes schema defined in Table 17, and the relations schema defined in Table 18.

Node Type	LOD relation	Property
Object	edm:isShownAt	shownAt
	title	title
	dc:DescriptionLangAware	description
	edm:Preview	img
creator	dc:Creator	creator_name
place	edm:PlaceLatitude	place_latitude
	edm:PlaceLongitude	place_longitude
	edm:ConceptPrefLabelLangAware	label
collection	edm:europeanaCollectionName	name_collection
concept	edmConcept	concept_label

Table 17: Graph schema for the background knowledge base

An illustrative subgraph is show in Figure 20.

Relation Type	Domain	Range
created_by	object	creator
located_at	object	place
in_collection	object	collection
has_concept	object	concept

Table 18: Relation schema for the background knowledge base

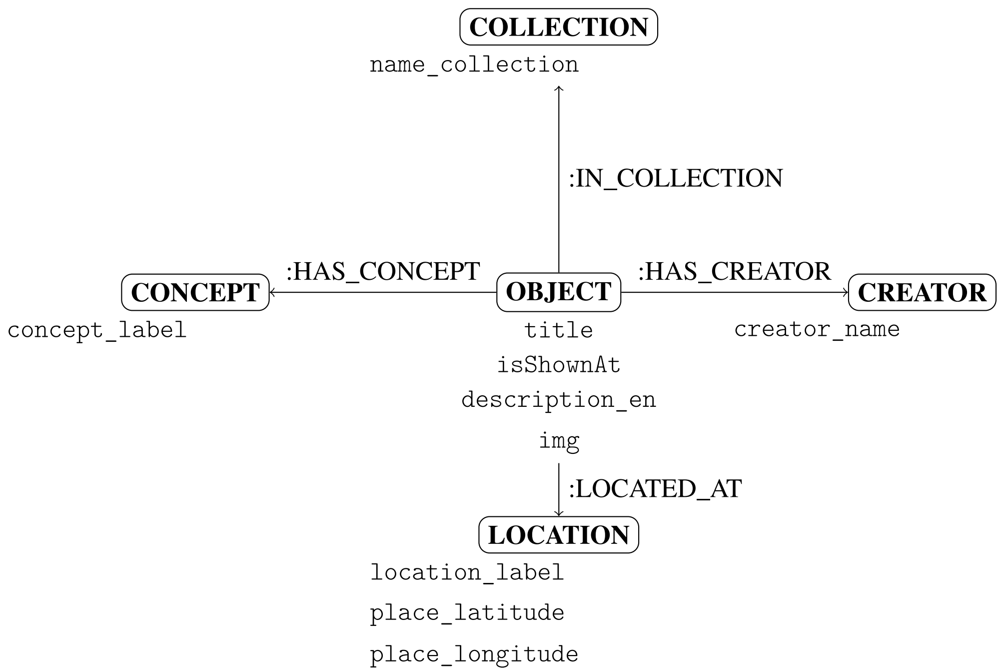


Figure 20: Illustrative subgraph extracted from the background knowledge graph

5.3.3 Terminology and Data Integration

Domain-specific knowledge is integrated by implementing terminologies for CH, created by employing well-established methodologies (di Buono et al., 2023; Nolano et al., 2022a, 2022b; Speranza et al., 2020; Speranza et al., 2021) to extract domain-specific terms representing entity mentions of interest.
These terms are used for the entity extraction step within the NLU phase. Furthermore, an additional step of taxonomy induction is implemented, so that specific entities referring to the same general concept would also be retrieved in case the

general concept is invoked by the user.
Two more lists of entities are employed, namely author names and place names (both extracted from the list of creators and places previously retrieved).

5.3.4 Natural Language Understanding

This component aims at obtaining a formal representation of users' natural language utterances. In order to do so, sentences are first classified into a set of labels, then entities are extracted based on domain-specific terminologies, and finally given specific dialogue information the question is parsed into a query in formal language. Since the KG used is implemented using neo4j, the formal language of choice is cypher.
This component is integrated using the Rasa NLU framework (Bocklisch et al., 2017), which makes use of a NLU pipeline every time a message is passed through the framework. While such a pipeline can be customized, the default configuration is implemented for MAGGIE.

The pipeline can be thought of as a linear sequence of consecutive components, taking a message as an input and giving an intent and entity classification as output. Through the pipeline, the message is featurized across several dimensions[16]. The default steps are:

1. WhitespaceTokenizer
2. RegexFeaturizer
3. LexicalSyntacticFeaturizer
4. CountVectorsFeaturizer
5. DIET Classifier

The first step in this pipeline is a tokenization based on white spaces, that returns a `token` object. Then, vectors are generated based on regular expression learnt at training time. In particular, such expressions mark whether certain features present in the training examples (e.g., presence of a numerical string, presence of an entity of a certain type) are also found in the input.
The following step is a lexical-syntactic featurizer that featurizes each token according to specific linguistic information (e.g., whether it appears at the beginning or at the end of a sentence, whether it is uppercase or lowercase, and part of speech tags).
Then, a vector of the input message is created based on token counts previously

[16]Implemented through sklearn. Read more at https://scikit-learn.org/stable/modules/generate d/sklearn.feature$_e$xtraction.text.CountVectorizer.html

learnt from the set of training examples.
Finally, the features are used in a Dual Intent and Entity Transformer (DIET) classifier, which makes use of a transformer shared across both tasks. At training stage, entity labels are learnt based on the transformer output sequence and a Conditional Random Field (CRF) on top of it.
For intent classification, instead, the transformer output is embedded together with intent labels into a single semantic vector space, and then the similarity between the correct label and the transformer output is maximized, while minimizing the similarity between the target label and negative samples.

Intent Classification

This module classifies user's intent into manually-defined, domain-specific labels. Such intents refer to the features desired for MAGGIE, in particular referring to:

1. Asking for items given a specific author;
2. asking for items given a specific entity type;
3. exploring a given location;
4. asking for pictures of items.

These features correspond, respectively, to the following intents:

1. `give_info_specific_author`;
2. `give_info_specific_type`;
3. `ask_for_pictures`;
4. `explore`.

Furthermore, general chatbot features are also accounted for by defining specific intentions (such as greeting, thanking and saying yes or no) which are trivially implemented, and as such will not be investigated.
Each intent is represented by a label and a series of example sentences. In defining such examples for intent classification, it is possible to generalize sentences so that certain spans of text can be represented by any entity mentions referring to a specific entity type. For instance, given the `explore` intent and the following example referring to it:

(8) What can I see in <u>Paris</u>?

The *Paris* span of text can be substituted by any other entity mention referring to a `city`. In RASA this is represented by the following syntax:

(9) What can I see in [Paris]{"entity" : "city"}?

Since such a representation improves the result for both intent classification and entity extraction, while also reducing the amount of actual examples needed for training, it is implemented for every intent in the course of this work.

give_info_specific_author This intent is implemented with examples such as:

- Show me something by [Paul Du Bois]{"entity" : "info_author_name"};
- I would like to see something by [Paul Du Bois]{"entity" : "info_author_name"};
- [Paul Du Bois]{"entity" : "info_author_name"}.

The case in which the whole sentence is represented by a single author's name is integrated so that the intent can be recognized even if the user is answering a question coming from the chatbot.

give_info_specific_type This intent is implemented with examples such as:

- I am interested in [churches]{"entity" : "info_entity_type"};
- I would like to see [paintings]{"entity" : "info_entity_type"}:
- [statues]{"entity" : "info_entity_type"}.

The intent example represented only by the entity type is included for the reasons explained above.

ask_for_pictures Users should be able to ask for pictures of either explicit items, or items previously mentioned in the dialogue. As such, the examples for this intent are the following:

- Show me some pictures;
- I would love to see some pictures of works by [René Magritte]{"entity" : "info_author_name"};
- Please, show me some pictures of [Paul Du Bois]{"entity" : "info_author_name"}'s masterpieces.

explore One of the main features of MAGGIE is the possibility for users to explore locations, either by explicitly stating which location they are interested in, or by enabling the use of geo-localized information. Examples for this intent reflect this features, and are as follows:

- What can I see around me?
- What can I see in [Paris]{"entity" : "city"}?
- Is there anything interesting to see around here?
- Is there anything interesting to see in [Paris]{"entity" : "city"}?

- [Paris]{"entity" : "city"}

In case no explicit location is stated in the utterance and geolocalization is enabled, a module is tasked with retrieving the geographical location of the users. In case no explicit location is stated, and geolocalization is not enabled, the chatbot asks for an explicit location.

Entity Extraction

While intents are manually defined, the list of entities to be recognized in sentences is extracted in the course of the previous steps of Terminology Integration and Data Alignment. In particular, entity mentions referring to the following concepts are extracted:

- entity types referring to domain-specific items that the users might be interested in (e.g., woodworking, statues, art exhibitions);
- author names (e.g., René Magritte, Pieter van Lint, Jean Boulaese);
- places name (e.g., Paris, Bruxelles, Royal Museum of Fine Arts of Belgium).

Since the representation employed in the EDM does not use a fine-grained set of relations to define an item's features (e.g., style, material, object type), but rather it defines such information under the umbrella relation `edmConcept`, the first set of entity mentions defines a set of features that can describe items across several dimensions. Author names, on the other hand, are implemented in order to give users the ability to search for specific authors and creators.

These entities are collected in separate *look-up tables*, one for each concept type, represented as a YAML file following the structure shown in Figure 21. Each

```
nlu
- lookup info\_author\_name
examples |
- Franceise Kraus
- Vincent Hensbergh
- Francisco Labata
...
```

Figure 21: Example of the YAML file used to represent authors.

lookup table has its own label, followed by a series of examples that the entity

extraction will look for in every sentence, together with the previously mentioned DIET classifier.
Once these entities are extracted, they are stored in the form of *slots* that can be easily extracted and *remembered* for the following steps.

Semantic Parsing

Once intents and entities are extracted from an input utterance, they are mapped to a semantic query used to retrieve the desider items. In particular, a process of explicit semantic parsing is implemented by making use of specific *actions*.
Actions in Rasa are defined as specific responses that the assistant should perform in certain situations. While actions can take many forms (e.g. message responses, forms, slot validation), the actual process of semantic parsing is implemented through *custom actions*, which are custom codes that get called once their action is activated. Since these codes can be represented by almost anything, they can be used to call Python-defined scripts to parse input utterances into formal representations.
Since the background knowledge base is implemented in a graph-based structure using neo4j, the formal language of choice is *cypher query language*[17] (CQL). A basic CQL query takes the form defined in Figure 22, with `m` and `n` representing nodes, and `r` representing a relation between them. In this case, the query returns all the pairs of nodes connected by the specific relation, and the relation itself.
Since neo4j implements data in the form of a labeled property graph, further

```
MATCH (n)-[r]->(m)
RETURN n,r,m
```

Figure 22: Example of a cypher query

information can be added to such a graph in order to filter the results and retrieve the desired items. This is done either by implementing a `WHERE` clause, or by explicitly defining the properties of the nodes to be matched. These methods are shown, respectively, in Figure 23 and Figure 24.
In order to integrate such a process of semantic parsing, intents are first mapped

[17] https://neo4j.com/developer/cypher/

```
MATCH (n)-[r]->(m)
WHERE n.title = 'The Castle of the Pyrenees'
RETURN n,r,m
```

Figure 23: Example of a WHERE clause

```
MATCH (n:OBJECT {title : 'The Castle of the Pyrenees'})-[r
 ]->(m)
RETURN n,r,m
```

Figure 24: Example of property filtering

to specific relations and properties, so that a query can be built once they are predicted. Such mappings are shown in Table 19. They can be combined in order for more complex queries to be implemented, for instance in case more than one relation or filter is expressed in the input utterance. While the first three intents

Intent	Query
give_info_specific_author	`MATCH (o:OBJECT)-[:CREATED_BY]->(c:CREATOR {creator_name : author_name}) RETURN o`
give_info_specific_type	`MATCH (o:OBJECT)-[:HAS_CONCEPT]->(c:CONCEPT {creator_label : concept_name}) RETURN o`
ask_for_pictures	`MATCH (o:OBJECT) RETURN o.img`
explore	`MATCH (o:OBJECT) -[r:LOCATED_AT]-> (p:PLACE)`

Table 19: Mapping of intents to query

always apply their filter in the same way, in the case of **explore**, a more complex implementation is needed, according to whether the location is explicitly stated, or retrieved from geolocalization.
In case the location is explicitly stated by the user, the value is looked for in the `location_label` property of the location node, and in case it is found the object connected to it is retrieved. In case the user allows for the use of geolocalized data, on the other hand, the query calculates the distance between the current position of the user that of all the objects in the KB, using the `place_latitude` and `place_longitude` of the location nodes and the `point.distance` function in

cypher[18].

5.3.5 Dialogue Management

Rasa makes use of several policies for Dialogue Management. In particular, for MAGGIE a *stories* policy is implemented. This policy makes use of training data to train the assistant's dialogue management to help generalize to unseen conversations.

Stories are defined as a conversation between an user and the assistant written in a flexible format, where user inputs are expressed as intents and entities, and assistant responses are expressed as given *actions* the assistant will take. Stories can be both implemented manually and be updated in the course of actual dialogues. Stories are stored in the form of YAML files, each with its own label and a set of steps. In the course of a conversation, the dialogue management module tries to match the current conversation with any of the stories available. If it can match them with a certain level of confidence, it will move the dialogue on according to the steps defined, otherwise it will stop.
In the course of MAGGIE's development, general stories, such as the one shown

```
- story explore with type
steps
- intent greet
- action utter_greet
- intent explore
- action utter_explore_ask_info
- intent give_info_specific_type
- action action_retrieve_explore
```

Figure 25: Example of a story

in Figure 25, are deployed. Such a story can generalize over a dialogue such as:

USER: Hi MAGGIE! [Intent: `greet`]

MAGGIE: Hi {username} [Action: `utter_greet`]

[18] https://neo4j.com/docs/cypher-manual/current/functions/spatial/

USER: What can I see around me? [Intent: `explore`]

MAGGIE: Are you interested in anything? [Action: `utter_explore_ask_info`]

USER: Yes, I am interested in paintings [Intent: `give_info_specific_type`]

MAGGIE: Let me think for a minute... [Action: `action_retrieve_explore`]

MAGGIE: Here are the paintings you can see in Brussels [Action: `action_retrieve_explore`]

MAGGIE: Do you want to see more? [Action: `action_ask_for_more`]

In the course of stories' development, new actions were added in order to give MAGGIE the ability to ask for more information in case they were not explicitly stated by the users in the course of dialogue. This is the case, for instance, of the `utter_explore_ask_info` action in Figure 25, which is a specific action that generate a question in case the action intention is evoked but not enough information about the desired items are present.

5.3.6 Answer Generation

Just as actions are used to map natural language utterances to cypher queries, they are also used to implement the generation of outputs in natural language from the conversational agent. As it is usually the case for NLP modules in customer service environments, these outputs are manually predefined (Kvale et al., 2020), and serve two main purposes: move the conversation further on, and answer questions. In order to move the conversation on, several utterances are defined so that MAGGIE can react to specific circumstances, such as being greeted, being thanked, or not having enough information to answer a query. Similarly, question prompts are also included, in order to add additional filters to a query (*Are you interested in anything?*) and to give more results (*Do you want to see more?*).
Then, once a list of entries is extracted given the semantic query, a maximum of three items from this list are shown as answers to the user, in the form of hyperlinks to the object's uri. Then, in case more items are available in the list, an action is implemented to ask whether the user wants to see more results, and in case they do, MAGGIE shows them further entries from the KG.
By implementing answers in the form of hyperlinks, the `shownAt` URI is integrated in the generated answers so that users can access further information about items, following Linked Open Data principles.

In case pictures are asked for by the user, they are shown in the form of a gallery making use of an image carousel, so that it is easy to display and navigate through the retrieved pictures, particularly using a mobile device.

Chapter 6

Conclusions

This work investigated several methodologies to integrate structured information with raw texts for Natural Language Understanding purposes.

Chapter 1 gives a general outline of the context in which this work lies, first by defining the general field and scopes of applications of natural language understanding, then moving on to natural language representation, in particular according to two main paradigms: text-based and graph-based. The differences between these two approaches have been underlined, in particular in relation to the issues present in purely distributional representations, generally used as the go-to methods for natural language processing purposes.
Chapter 2 offers a literature review of topics related to natural language representation, once again with regards to two main paradigms: neural representations and symbolic representations. A great deal of attention has been put in underlining the efforts in combining these two approaches in order to solve issues inherently present in both.

The second part of the work describes several contributions made in the field of NLU applications, in particular in relation to the extraction and reuse of semantic knowledge from unstructured texts, in the form of graph-like structures.
Chapter 3, in particular, describes two methodologies proposed to extract semantic information from domain-specific texts, by a combination of several sources of knowledge, namely NLP tools, ontologies and knowledge graphs. The presented results show the benefits of such integration, in particular in relation to tasks such as named entity recognition and keyword extraction.
While the results provided for the NEAT methodology represent a reliable source

for domain-specific NEs, both the process of semantic projection and entity validation could be improved to guarantee a higher precision regarding the proposed annotations. For instance, the exploitation of semantic information could be beneficial in solving ambiguities and other borderline cases.
For instance, while *arco ribassato* (segmental arch) and *arco lungo* (longbow) are described by the same pattern in Italian, namely *arco* + ADJ, they refer to two completely different concepts. This ambiguity is impossible to solve by simply looking at the surface form of the head (because of the homonymy in Italian) and the possible syntactic pattern, but could be tackled through the implementation of a measurement of the correlation between concepts' representations and specific set of words.
Another way to improve the results could be through the integration of transformation rules in order to move from the specific token-level structures to more general phrase-level structures.
Once the methodology is honed in this sense, it can be tested for downstream tasks, for instance it could be used in handling specific terminology for Machine Translation systems, Entity Linking and Word Sense Disambiguation.

Regarding the multiword-based methodology, the two experiments show meaningful results in the field of domain-specific semantic enrichment. Nevertheless, the proposed processes is to be refined in order to improve the results. In particular, a domain-specific BERT-like embedding model, and a more refined method for calculating the MWE embeddings might improve the concept discovery stage. Furthermore, a more flexible and lexically grounded set of linguistic patterns might help improve current results, while also reducing the noise currently present in the resource.

Chapter 4 shows an evaluation made on relation extraction models to test the integration of external, structured information in this task. In particular, the methodology used makes use of an adversarial environment to check how robust the models are, and how much they are affected by noise in the input data. The results shows that, while graph-based relation extraction shows promising results in the standard evaluation framework, it is still heavily affected by permutations in testing data.
The experiment described in relation to this work show that the introduction of structured data from a background source of knowledge is particularly useful for graph-based representations for RE. In particular, it is of notice that the combination of semantic and syntactic dependencies leads to an improvement of up to 43% in F1 over the original graph-based baseline. This is further improved by the addition of features extracted from Wikidata, which leads to a further 33%

improvement in the F1 score.
Regarding the robustness of models to adversarial attacks, while the original hypothesis was that graph-based representations for RE would lead to an improvement in this aspect, the experiments prove that GNN show worse results both on the original RE task, while also being more affected by adversarial permutations. Furthermore, the experiments show that certain patterns arise across different models when they face specific permutations, proving that specific changes in data have the same effects on results, regardless of the employed model.

Chapter 5 describes two approaches to question answering over knowledge graph, the first one based on a grammar generation architecture implemented using linked open data and language-specific patterns, and the second one based on an explicit semantic parsing step over a labeled property graph through intent classification and entity extraction.
In particular, while both methodologies share the use of graph-based knowledge bases as source of knowledge for the answers, their differences in terms of practical implementations are described.

The methodologies presented for the implementation of graph-based structures in natural language understanding are shown and implemented in practical applications, which highlight benefits and shortcomings of the processes. With regards to research questions, **RQ1** was interested in whether graph-based knowledge integration can show any benefits in NLU-related tasks. It has been shown that knowledge graphs and ontologies can be used to improve results for specific NLP tasks, as it is the case for domain-specific semantic enrichment, and, with regards to relation extraction it shows a good degree of possibilities with regards to knowledge injection.
RQ2, on the other hand, focuses on the type of data format through which graphs can be implemented for NLU. In this work it has been shown that both RDF-based KBs and LPGs can be used to great extent, and that mapping one to the other and back can be useful in solving the issues present in one by leveraging on the benefits of the other.
More specifically, Chapter 3 showed the benefits of integrating LOD in semantic enrichment using grounded RDF knowledge bases. Chapter 4, similarly, described the integration of LPGs in the task of relation extraction, while also showing the usefulness of rdf-based data injection for the task. Chapter 5, finally, showed the implementation of both RDF knowledge bases and LPGs for the task of QA, and Section 5.3, in particular, proved the process of mapping RDF graphs to LPGs to increase performances in a real-world environment.

This thesis highlights the recent interests in themes relating the integration of structured knowledge together with distributional representations in order to solve many of the issues still left unsolved. Furthermore, it focuses on how some of the issues can be solved by leveraging graph-based representations for texts and background knowledge. The experiments described in this work are only some of the many applications which have been proposed in recent years and that continue to be proposed still. In future works, such experiments should be improved and expanded upon, in order to have more detailed results and better methodologies.

Bibliography

Abiteboul, S. (1997). Querying semi-structured data. In F. N. Afrati & P. G. Kolaitis (Eds.), *Database theory - ICDT '97, 6th international conference, delphi, greece, january 8-10, 1997, proceedings* (pp. 1–18). Springer. https://doi.org/10.1007/3-540-62222-5_33

Acosta, J., Lamaute, N., Luo, M., Finkelstein, E., & Andreea, C. (2017). Sentiment analysis of twitter messages using word2vec. *Proceedings of Student-Faculty Research Day, CSIS, Pace University*, *7*, 1–7.

Adi, Y., Kermany, E., Belinkov, Y., Lavi, O., & Goldberg, Y. (2017). Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. https://openreview.net/forum?id=BJh6Ztuxl

Ahn, S., Choi, H., Pärnamaa, T., & Bengio, Y. (2017). A neural knowledge language model.

Al-Ghezi, R., & Kurimo, M. Graph-based Syntactic Word Embeddings. English. In: Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs); 2020, 7. ISBN: 978-1-952148-42-2. http://urn.fi/URN:NBN:fi:aalto-202102091962

Ali, W., Saleem, M., Yao, B., Hogan, A., & Ngomo, A.-C. N. (2021). A survey of rdf stores & sparql engines for querying knowledge graphs. *The VLDB Journal*, 1–26.

Allen, J. (1987). *Natural language understanding*. Benjamin-Cummings Publishing Co., Inc.

Alt, C., Gabryszak, A., & Hennig, L. (2020). Probing linguistic features of sentence-level representations in neural relation extraction. *arXiv preprint arXiv:2004.08134*.

Altszyler, E., Sigman, M., & Slezak, D. F. (2016). Comparative study of LSA vs word2vec embeddings in small corpora: A case study in dreams database. *CoRR*, *abs/1610.01520*. http://arxiv.org/abs/1610.01520

Angles, R. (2018). The property graph database model. *AMW*.

Angles, R., & Gutierrez, C. (2008). Survey of graph database models. *ACM Comput. Surv.*, *40*(1). https://doi.org/10.1145/1322432.1322433

Angles, R., Thakkar, H., & Tomaszuk, D. (2019). Rdf and property graphs interoperability: Status and issues. *AMW*.

Artzi, Y., & Zettlemoyer, L. (2011). Bootstrapping semantic parsers from conversations. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 421–432. https://aclanthology.org/D11-1039

Artzi, Y., & Zettlemoyer, L. (2013). Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics*, *1*, 49–62. https://doi.org/10.1162/tacl_a_00209

Bach, N., & Badaskar, S. (2007). A review of relation extraction. *Literature review for Language and Statistics II*, *2*, 1–15.

Bader, S., & Hitzler, P. (2005). Dimensions of neural-symbolic integration - a structured survey.

Bai, J., Wang, Y., Chen, Y., Yang, Y., Bai, J., Yu, J., & Tong, Y. (2021). Syntax-BERT: Improving pre-trained transformers with syntax trees. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 3011–3020. https://doi.org/10.18653/v1/2021.eacl-main.262

Baker, C. F., Fillmore, C. J., & Lowe, J. B. The Berkeley FrameNet project. In: *Coling-acl '98: Proceedings of the conference*. Montreal, Canada, 1998, 86–90.

Baker, M. (1988). *Incorporation: A theory of grammatical function changing*. Chicago: University of Chicago Press.

Baldini Soares, L., FitzGerald, N., Ling, J., & Kwiatkowski, T. (2019). Matching the blanks: Distributional similarity for relation learning. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2895–2905. https://doi.org/10.18653/v1/P19-1279

Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., & Schneider, N. (2013). Abstract Meaning Representation for sembanking. *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, 178–186. https://aclanthology.org/W13-2322

Bao, J., Duan, N., Yan, Z., Zhou, M., & Zhao, T. (2016). Constraint-based question answering with knowledge graph. *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2503–2514. https://aclanthology.org/C16-1236

Bastings, J., Titov, I., Aziz, W., Marcheggiani, D., & Sima'an, K. (2017). Graph convolutional encoders for syntax-aware neural machine translation. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1957–1967. https://doi.org/10.18653/v1/D17-1209

Bauer, F., & Kaltenböck, M. (2011). Linked open data: The essentials. *Edition mono/monochrom, Vienna*, *710*, 21.

Baumgartner, M., Zhang, W., Paudel, B., Dell'Aglio, D., Chen, H., & Bernstein, A. (2018). Aligning knowledge base and document embedding models using regularized multi-task learning: 17th international semantic web conference, monterey, ca, usa, october 8–12, 2018, proceedings, part i. https://doi.org/10.1007/978-3-030-00671-6_2

Belinkov, Y., & Bisk, Y. (2017). Synthetic and natural noise both break neural machine translation. https://doi.org/10.48550/ARXIV.1711.02173

Belinkov, Y., Gehrmann, S., & Pavlick, E. (2020). Interpretability and analysis in neural nlp. *Proceedings of the 58th annual meeting of the association for computational linguistics: tutorial abstracts*, 1–5.

Beltagy, I., Lo, K., & Cohan, A. (2019). SciBERT: A pretrained language model for scientific text. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3615–3620. https://doi.org/10.18653/v1/D19-1371

Bender, E. M., Flickinger, D., Oepen, S., Packard, W., & Copestake, A. (2015). Layers of interpretation: On grammar and compositionality. *Proceedings of the 11th International Conference on Computational Semantics*, 239–249. https://aclanthology.org/W15-0128

Bender, E. M., & Koller, A. (2020). Climbing towards NLU: On meaning, form, and understanding in the age of data. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5185–5198. https://doi.org/10.18653/v1/2020.acl-main.463

Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, 610–623.

Bengio, Y., Frasconi, P., & Simard, P. Y. (1993). The problem of learning long-term dependencies in recurrent networks. *IEEE International Conference on Neural Networks*, 1183–1188 vol.3.

Bengio, Y., Simard, P. Y., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, *5 2*, 157–66.

Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A neural probabilistic language model. *J. Mach. Learn. Res.*, *3*, 1137–1155.

Benikova, D., Biemann, C., & Reznicek, M. (2014). NoSta-D named entity annotation for German: Guidelines and dataset. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 2524–2531. http://www.lrec-conf.org/proceedings/lrec2014/pdf/276_Paper.pdf

Benz, V., Cimiano, P., Elahi, M. F., & Ell, B. (2020). Generating grammars from lemon lexica for questions answering over linked data: A preliminary analysis. *NLIWOD workshop at ISWC*, *2722*, 40–55.

Berant, J., Chou, A. K., Frostig, R., & Liang, P. (2013). Semantic parsing on freebase from question-answer pairs. *EMNLP*.

Bhatt, G., Bansal, H., Singh, R., & Agarwal, S. (2020). How much complexity does an rnn architecture need to learn syntax-sensitive dependencies? *arXiv preprint arXiv:2005.08199*.

Bian, J., Gao, B., & Liu, T.-Y. (2014). Knowledge-powered deep learning for word embedding. https://www.microsoft.com/en-us/research/publication/knowledge-powered-deep-learning-for-word-embedding/

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, *3*(null), 993–1022.

Block, N. (1981). Psychologism and behaviorism. *Philosophical Review*, *90*(1), 5–43. https://doi.org/10.2307/2184371

Bocklisch, T., Faulkner, J., Pawlowski, N., & Nichol, A. (2017). Rasa: Open source language understanding and dialogue management. *arXiv preprint arXiv:1712.05181*.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146.

Bollacker, K., Tufts, P., Pierce, T., & Cook, R. (2007). A platform for scalable, collaborative, structured information integration. *Intl. Workshop on Information Integration on the Web (IIWeb'07)*, 22–27.

Bordes, A., Chopra, S., & Weston, J. (2014). Question answering with subgraph embeddings. https://doi.org/10.48550/ARXIV.1406.3676

Bordes, A., Weston, J., & Usunier, N. (2014). Open question answering with weakly supervised embedding models. *CoRR*, *abs/1404.4326*. http://arxiv.org/abs/1404.4326

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf

Bordes, A., Usunier, N., Chopra, S., & Weston, J. (2015). Large-scale simple question answering with memory networks. *CoRR*, *abs/1506.02075*. http://arxiv.org/abs/1506.02075

Boudin, F. (2018). Unsupervised keyphrase extraction with multipartite graphs. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 667–672. https://doi.org/10.18653/v1/N18-2105

Bougouin, A., Boudin, F., & Daille, B. (2013). TopicRank: Graph-based topic ranking for keyphrase extraction. *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, 543–551. https://aclanthology.org/I13-1062

Bouraoui, Z., Camacho-Collados, J., & Schockaert, S. (2019). Inducing relational knowledge from BERT. *CoRR*, *abs/1911.12753*. http://arxiv.org/abs/1911.12753

Brandom, R. (2010). Inferentialism and some of its challenges. *Reading brandom* (pp. 169–190). Routledge.

Brandom, R. B. (2000). *Articulating reasons: An introduction to inferentialism*. Harvard University Press.

Brandsen, A., Verberne, S., Wansleeben, M., & Lambers, K. (2020). Creating a dataset for named entity recognition in the archaeology domain. *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 4573–4577. https://aclanthology.org/2020.lrec-1.562

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine [Proceedings of the Seventh International World Wide Web Conference]. *Computer Networks and ISDN Systems*, *30*(1), 107–117. https://doi.org/https://doi.org/10.1016/S0169-7552(98)00110-X

Brown, S. W., Bonial, C., Obrst, L., & Palmer, M. (2017). The rich event ontology. *Proceedings of the Events and Stories in the News Workshop*, 87–97.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., … Amodei, D. (2020). Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (pp. 1877–1901). Curran Associates, Inc. https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

Bullinaria, J., & Levy, J. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study: A computational study. *BEHAVIOR RESEARCH METHODS*, *39*(3), 510–526. https://doi.org/10.3758/BF03193020

Burge, T. (1986). On davidson's 'saying that'. In E. LePore (Ed.), *Truth and interpretation: Perspectives on the philosophy of donald davidson*. Blackwell.

Busbridge, D., Sherburn, D., Cavallo, P., & Hammerla, N. Y. (2019). Relational graph attention networks. https://doi.org/10.48550/ARXIV.1904.05811

Callahan, A., Cruz-Toledo, J., Ansell, P., & Dumontier, M. (2013). Bio2rdf release 2: Improved coverage, interoperability and provenance of life science linked data. *7882*, 200–212. https://doi.org/10.1007/978-3-642-38288-8_14

Carnap, R. (1947). *Meaning and necessity: A study in semantics and modal logic*. Chicago, IL, USA: University of Chicago Press.

Carpenter, B. (1997). *Type-logical semantics*. MIT Press.

Caruana, R. (1993). Multitask learning: A knowledge-based source of inductive bias. *Proceedings of the Tenth International Conference on Machine Learning*, 41–48.

Celikyilmaz, A., Hakkani-Tür, D., Pasupat, P., & Sarikaya, R. (2015). Enriching word embeddings using knowledge graph for semantic tagging in conversational dialog systems. https://www.microsoft.com/en-us/research/publication/enriching-word-embeddings-using-knowledge-graph-for-semantic-tagging-in-conversational-dialog-systems/

Chakraborty, N., Lukovnikov, D., Maheshwari, G., Trivedi, P., Lehmann, J., & Fischer, A. (2019). Introduction to neural network based approaches for question answering over knowledge graphs. *arXiv preprint arXiv:1907.09361*.

Chang, W.-C., Yu, H.-F., Zhong, K., Yang, Y., & Dhillon, I. (2019). X-bert: Extreme multi-label text classification with using bidirectional encoder representations from transformers. *arXiv preprint arXiv:1905.02331*.

Chen, H., Huang, S., Chiang, D., & Chen, J. (2017). Improved neural machine translation with a syntax-aware encoder and decoder. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1936–1945. https://doi.org/10.18653/v1/P17-1177

Chen, K., Yang, M., Utiyama, M., Sumita, E., Wang, R., & Zhang, M. (2022). Effective graph context representation for document-level machine translation [Main Track]. In L. D. Raedt (Ed.), *Proceedings of the thirty-first international joint conference on artificial intelligence, IJCAI-22* (pp. 4079–4085). International Joint Conferences on Artificial Intelligence Organization. https://doi.org/10.24963/ijcai.2022/566

Cheng, J., Reddy, S., Saraswat, V., & Lapata, M. (2017). Learning structured natural language representations for semantic parsing. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 44–55. https://doi.org/10.18653/v1/P17-1005

Chiarcos, C., & Fäth, C. (2017). Conll-rdf: Linked corpora done in an nlp-friendly way. In J. Gracia, F. Bond, J. P. McCrae, P. Buitelaar, C. Chiarcos, & S. Hellmann (Eds.), *Language, data, and knowledge - first international conference, LDK 2017, galway, ireland, june 19-20, 2017, proceedings* (pp. 74–88). Springer. https://doi.org/10.1007/978-3-319-59888-8_6

Chiarcos, C., & Glaser, L. (2020). A tree extension for conll-rdf. In N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the 12th language resources and evaluation conference, LREC 2020, marseille, france, may 11-16, 2020* (pp. 7161–7169). European Language Resources Association. https://aclanthology.org/2020.lrec-1.885/

Chiarcos, C., Ionov, M., Glaser, L., & Fäth, C. (2021). An ontology for conll-rdf: Formal data structures for TSV formats in language technology. In D. Gromann, G. Sérasset, T. Declerck, J. P. McCrae, J. Gracia, J. Bosque-Gil, F. Bobillo, & B. Heinisch (Eds.), *3rd conference on language, data and knowledge, LDK 2021, september 1-3, 2021, zaragoza, spain* (20:1–20:14). Schloss Dagstuhl - Leibniz-Zentrum für Informatik. https://doi.org/10.4230/OASIcs.LDK.2021.20

Chiticariu, L., Li, Y., & Reiss, F. (2013). Rule-based information extraction is dead! long live rule-based information extraction systems! *Proceedings of the 2013 conference on empirical methods in natural language processing*, 827–832.

Chiu, J. P., & Nichols, E. (2016). Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, *4*, 357–370.

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on Information Theory*, *2*(3), 113–124. https://doi.org/10.1109/TIT.1956.1056813

Chomsky, N. (1957). *Syntactic structures*. Mouton; Co.

Christopoulou, F., Miwa, M., & Ananiadou, S. (2019). Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 4925–4936. https://doi.org/10.18653/v1/D19-1498

Cimiano, P., Unger, C., & McCrae, J. (2014). Ontology-based interpretation of natural language. *Synthesis Lectures on Human Language Technologies*, *7*(2), 1–178. https://doi.org/10.2200/S00561ED1V01Y201401HLT024

Clark, S., & Pulman, S. G. (2007). Combining symbolic and distributional models of meaning. *Quantum Interaction*.

Clarke, J., Goldwasser, D., Chang, M.-W., & Roth, D. (2010). Driving semantic parsing from the world's response. *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, 18–27. https://aclanthology.org/W10-2903

Clausi, D. A., & Jernigan, M. E. (1998). A fast method to determine co-occurrence texture features. *IEEE Transactions on Geoscience and Remote Sensing*, *36*(1), 298–300.

Clinchant, S., Jung, K. W., & Nikoulina, V. (2019). On the use of BERT for neural machine translation. *Proceedings of the 3rd Workshop on Neural Generation and Translation*, 108–117. https://doi.org/10.18653/v1/D19-5611

Coden, A. R., Pakhomov, S. V., Ando, R. K., Duffy, P. H., & Chute, C. G. (2005a). Domain-specific language models and lexicons for tagging. *J. of Biomedical Informatics*, *38*(6), 422–430. https://doi.org/10.1016/j.jbi.2005.02.009

Coden, A. R., Pakhomov, S. V., Ando, R. K., Duffy, P. H., & Chute, C. G. (2005b). Domain-specific language models and lexicons for tagging. *Journal of Biomedical Informatics*, *38*(6), 422–430. https://doi.org/https://doi.org/10.1016/j.jbi.2005.02.009

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, *12*(null), 2493–2537.

Crespo, J., Barber, R., & Mozos, O. (2017). Relational model for robotic semantic navigation in indoor environments. *Journal of Intelligent & Robotic Systems*, *86*(3), 617–639.

Crespo, J., Castillo, J. C., Mozos, O. M., & Barber, R. (2020). Semantic information for robot navigation: A survey. *Applied Sciences*, *10*(2). https://doi.org/10.3390/app10020497

Croce, D., Zelenanska, A., & Basili, R. (2018). Neural learning for question answering in italian. *AI*IA*, *11298*, 389–402.

Cui, L., Wu, Y., Liu, S., & Zhang, Y. (2021). Knowledge enhanced fine-tuning for better handling unseen entities in dialogue generation. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2328–2337. https://doi.org/10.18653/v1/2021.emnlp-main.179

Culotta, A., & Sorensen, J. (2004). Dependency tree kernels for relation extraction. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, 423–429. https://doi.org/10.3115/1218955.1219009

Dai, A. M., & Le, Q. V. (2015). Semi-supervised sequence learning. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2015/file/7137debd45ae4d0ab9aa953017286b20-Paper.pdf

Dai, Q., Kong, F., & Dai, Q. (2019). Event temporal relation classification based on graph convolutional networks. https://doi.org/10.1007/978-3-030-32236-6_35

Das, A., Ganguly, D., & Garain, U. (2017). Named entity recognition with word embeddings and wikipedia categories for a low-resource language. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, *16*(3). https://doi.org/10.1145/3015467

Dauphin, Y. N., Fan, A., Auli, M., & Grangier, D. (2017). Language modeling with gated convolutional networks. *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, 933–941.

David, Y., Grace, N., Richard, W., et al. (2001). Inducing multilingual text analysis tools via robust projection across aligned corpora. *Proceedings of the First International Conference on Human Language Technology Research*, 1–8.

Davidson, D. (1967). Truth and meaning. *Synthese*, *17*(1), 304–323. https://doi.org/10.1007/bf00485035

Davidson, D. (1976). Reply to foster. In G. Evans & J. H. McDowell (Eds.), *Truth and meaning: Essays in semantics* (pp. 33–41). Clarendon Press.

Dayhoff, J. E., & DeLeo, J. M. (2001). Artificial neural networks: Opening the black box. *Cancer: Interdisciplinary International Journal of the American Cancer Society*, *91*(S8), 1615–1635.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science 41*, 391–407.

Denton, W. (2003). How to make a faceted classification and put it on the web. *Retrieved May*, *3*, 2006.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, & T. Solorio (Eds.), *Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: Human language technologies, NAACL-HLT 2019, min-*

neapolis, mn, usa, june 2-7, 2019, volume 1 (long and short papers) (pp. 4171–4186). Association for Computational Linguistics. https://doi.org/10.18653/v1/n19-1423

de Vries, G. K. (2013). A fast approximation of the weisfeiler-lehman graph kernel for rdf data. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 606–621.

di Buono, M. P., Gonçalo Oliveira, H., Barbu Mititelu, V., Spahiu, B., & Nolano, G. (2022). Paving the way for enriched metadata of linguistic linked data. *Semantic Web*, (Preprint), 1–25.

di Buono, M. P., Nolano, G., & Monti, J. (2023). Neat - named entities in archaeological texts: A semantic approach to term extraction and classification. *Digital Scholarship in the Humanities*.

di Buono, M. P. (2017). An ontology-based method for extracting and classifying domain-specific compositional nominal compounds. *Proceedings of the Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, 83–88. https://doi.org/10.18653/v1/W17-2211

Dimitrakis, E., Sgontzos, K., & Tzitzikas, Y. (2019). A survey on question answering systems over linked data and documents. *Journal of Intelligent Information Systems*, *55*, 233–259.

Ding, L., & Finin, T. (2006). Characterizing the semantic web on the web. In I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, & L. M. Aroyo (Eds.), *The semantic web - iswc 2006* (pp. 242–257). Springer Berlin Heidelberg.

Doerr, M. (2003). The cidoc conceptual reference module: An ontological approach to semantic interoperability of metadata. *AI Mag.*, *24*(3), 75–92.

Dogan, C., Dutra, A., Gara, A., Gemma, A., Shi, L., Sigamani, M., & Walters, E. (2019). Fine-grained named entity recognition using elmo and wikidata. *arXiv preprint arXiv:1904.10503*.

Dong, L., Wei, F., Zhou, M., & Xu, K. (2015). Question answering over Freebase with multi-column convolutional neural networks. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 260–269. https://doi.org/10.3115/v1/P15-1026

Dong, L., & Lapata, M. (2016). Language to logical form with neural attention. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 33–43. https://doi.org/10.18653/v1/P16-1004

Dostert, B., & Thompson, F. B. (1969). Rel: A rapidly extensible language system ii. rel english. *Proceedings of the 1969 Conference on Computational Linguistics*, 1–38. https://doi.org/10.3115/990403.990438

Dowty, D. (1991). Thematic proto-roles and argument selection. *Language*, *67*(3), 547–619. Retrieved October 24, 2022, from http://www.jstor.org/stable/415037

Dumais, S. T., Furnas, G. W., Landauer, T. K., Deerwester, S. C., & Harshman, R. A. (1988). Using latent semantic analysis to improve access to textual information. *CHI*, 281–285.

Ebrahimi, J., Rao, A., Lowd, D., & Dou, D. (2018). HotFlip: White-box adversarial examples for text classification. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 31–36. https://doi.org/10.18653/v1/P18-2006

Edouard, A., Cabrio, E., Tonelli, S., & Le-Thanh, N. (2017). Graph-based event extraction from Twitter. *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, 222–230. https://doi.org/10.26615/978-954-452-049-6_031

Eger, S., Şahin, G. G., Rücklé, A., Lee, J.-U., Schulz, C., Mesgar, M., Swarnkar, K., Simpson, E., & Gurevych, I. (2019). Text processing like humans do: Visually attacking and shielding NLP systems. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 1634–1647. https://doi.org/10.18653/v1/N19-1165

Einolghozati, A., Pasupat, P., Gupta, S., Shah, R., Mohit, M., Lewis, M., & Zettlemoyer, L. (2019). Improving semantic parsing for task oriented dialog. *arXiv preprint arXiv:1902.06000*.

Ekštein, K., & Krčmář, L. (2013). Automatic lsa-based retrieval of synonyms (for search space extension). In F. L. Gaol (Ed.), *Recent progress in data engineering and internet technology* (pp. 79–85). Springer Berlin Heidelberg.

Elahi, M. F., Ell, B., Grimm, F., & Cimiano, P. (2021). Question answering on rdf data based on grammars automatically generated from lemon models. *SEMANTiCS Posters&Demos*.

Elahi, M. F., Nolano, G., Grimm, F., & Röhler, A. (2021). Hackathon on question answering based on automatically generated grammars (5-9 july 2021). https://scdemo.techfak.uni-bielefeld.de/qahackathon/tutorial/abstract.pdf

Elfes, A. (1987). Sonar-based real-world mapping and navigation. *IEEE Journal on Robotics and Automation*, *3*(3), 249–265. https://doi.org/10.1109/JRA.1987.1087096

Elman, J. L. (1990). Finding structure in time. *COGNITIVE SCIENCE*, *14*(2), 179–211.

Elsahar, H., Vougiouklis, P., Remaci, A., Gravier, C., Hare, J., Laforest, F., & Simperl, E. (2018). T-REx: A large scale alignment of natural language with knowledge base triples. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. https://aclanthology.org/L18-1544

Emerson, G. (2020). What are the goals of distributional semantics? *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7436–7453. https://doi.org/10.18653/v1/2020.acl-main.663

Eriguchi, A., Hashimoto, K., & Tsuruoka, Y. (2016). Tree-to-sequence attentional neural machine translation. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 823–833. https://doi.org/10.18653/v1/P16-1078

Faber, P., & L'Homme, M.-C. (2014). Lexical semantic approaches to terminology. *Terminology*, *20*(2), 143–150.

Fagan, S., & Gençay, R. (2010). An introduction to textual econometrics. *Handbook of empirical economics and finance*, 133–154. https://doi.org/10.1201/b10440-6

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., & Smith, N. A. (2015). Retrofitting word vectors to semantic lexicons. *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1606–1615. https://doi.org/10.3115/v1/N15-1184

Feder, A., Keith, K. A., Manzoor, E., Pryzant, R., Sridhar, D., Wood-Doughty, Z., Eisenstein, J., Grimmer, J., Reichart, R., Roberts, M. E., Stewart, B. M., Veitch, V., & Yang, D. (2022). Causal Inference in Natural Language Processing: Estimation, Prediction, Interpretation and Beyond. *Transactions of the Association for Computational Linguistics*, *10*, 1138–1158. https://doi.org/10.1162/tacl_a_00511

Felicetti, A., Galluccio, I., Luddi, C., Mancinelli, M. L., Scarselli, T., & Madonna, A. D. (2015). Integrating terminological tools and semantic archaeological information: The iccd ra schema and thesaurus. *EMF-CRM@ TPDL*, 28–43.

Ferrone, L., & Zanzotto, F. M. (2020). Symbolic, distributed, and distributional representations for natural language processing in the era of deep learning: A survey. *Frontiers in Robotics and AI*, *6*, 153.

Fillmore, C. J. (1968). The case for case. In E. W. Bach & R. T. Harms (Eds.), *Universals in linguistic theory* (pp. 1–88). Holt, Rinehart & Winston.

Fillmore, C. J. (1977). The need for a frame semantics in linguistics. In H. Karlgren (Ed.), *Statistical methods in linguistics*. Scriptor.

Firoozeh, N., Nazarenko, A., Alizon, F., & Daille, B. (2020). Keyword extraction: Issues and methods. *Natural Language Engineering*, *26*(3), 259–291. https://doi.org/10.1017/S1351324919000457

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

Florescu, C., & Caragea, C. (2017). PositionRank: An unsupervised approach to keyphrase extraction from scholarly documents. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1105–1115. https://doi.org/10.18653/v1/P17-1102

Forbes, M., Holtzman, A., & Choi, Y. (2019). Do neural language representations learn physical commonsense? *Proceedings of the 41st Annual Conference of the Cognitive Science Society*.

Freeman, L. (1977). A set of measures of centrality based on betweenness. *Sociometry*, *40*, 35–41. https://doi.org/10.2307/3033543

Frege, G. (1948). Sense and reference. *Philosophical Review*, *57*(3), 209–230. https://doi.org/10.2307/2181485

Fung, P., & Chen, B. (2004). Biframenet: Bilingual frame semantics resource construction by cross-lingual induction. *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, 931–937.

Gangemi, A., Alam, M., Asprino, L., Presutti, V., & Recupero, D. R. (2016). Framester: A wide coverage linguistic linked data hub. *European knowledge acquisition workshop*, 239–254.

Gao, T., Han, X., Zhu, H., Liu, Z., Li, P., Sun, M., & Zhou, J. (2019). FewRel 2.0: Towards more challenging few-shot relation classification. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 6250–6255. https://doi.org/10.18653/v1/D19-1649

Gardner, M., & Krishnamurthy, J. (2016). Open-vocabulary semantic parsing with both distributional statistics and formal knowledge. *AAAI Conference on Artificial Intelligence*.

Garg, S., & Ramakrishnan, G. (2020). BAE: BERT-based adversarial examples for text classification. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6174–6181. https://doi.org/10.18653/v1/2020.emnlp-main.498

Giannesini, F., Kanoui, H., Pasero, R., & Van Caneghem, M. (1986). *Prolog*. Addison-Wesley Longman Publishing Co., Inc.

Gibson, E. (1998). Linguistic complexity: Locality of syntactic dependencies. *Cognition*, *68*(1), 1–76. https://doi.org/https://doi.org/10.1016/S0010-0277(98)00034-1

Gibson, E. (2000). The dependency locality theory: A distance-based theory of linguistic complexity. *Image, Language, Brain: Papers from the First Mind Articulation Project Symposium*.

Goldberg, Y. (2019). Assessing bert's syntactic abilities. *CoRR*, *abs/1901.05287*. http://arxiv.org/abs/1901.05287

Goldwasser, D., & Roth, D. (2011). Learning from natural instructions. *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, 1794–1800. http://dan-goldwasser.com/publications/ijcai2011.pdf

Golfarelli, M., Maio, D., & Rizzi, S. (2001). Correction of dead-reckoning errors in map building for mobile robots. *Robotics and Automation, IEEE Transactions on*, *17*, 37–47. https://doi.org/10.1109/70.917081

Gong, C., He, D., Tan, X., Qin, T., Wang, L., & Liu, T.-Y. (2018). Frage: Frequency-agnostic word representation. *Advances in neural information processing systems*, *31*.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In Y. Bengio & Y. LeCun (Eds.), *3rd international conference on learning representations, ICLR 2015, san diego, ca, usa, may 7-9, 2015, conference track proceedings*. http://arxiv.org/abs/1412.6572

Goyal, P., & Ferrara, E. (2018). Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, *151*, 78–94.

Gozalo-Brizuela, R., & Garrido-Merchan, E. C. (2023). Chatgpt is not all you need. a state of the art review of large generative ai models. *arXiv preprint arXiv:2301.04655*.

Gracia, J., & Mena, E. (2012). Semantic heterogeneity issues on the web. *IEEE Internet Computing*, *16*(5), 60–67.

Green, B., Wolf, A., Chomsky, C., & Laughery, K. (1961). Baseball:an automatic question-answerer. *IRE-AIEE-ACM Computer Conference (Western)*.

Greenberg, J. H. (1949). The logical analysis of kinship. *Philosophy of Science*, *16*(1), 58–64. https://doi.org/10.1086/287012

Grice, H. P. (1989). *Studies in the way of words*. Cambridge: Harvard University Press.

Groschwitz, J., Koller, A., & Teichmann, C. (2015). Graph parsing with s-graph grammars. *ACL*.

Grover, A., & Leskovec, J. (2016). Node2vec: Scalable feature learning for networks. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Gruber, J. S. (1965). *Studies in lexical relations* (Doctoral dissertation). MIT. Cambridge, MA.

Guan, J., Huang, F., Zhao, Z., Zhu, X., & Huang, M. (2020). A Knowledge-Enhanced Pretraining Model for Commonsense Story Generation. *Transactions of the Association for Computational Linguistics*, *8*, 93–108. https://doi.org/10.1162/tacl_a_00302

Gul, F., Rahiman, W., & Nazli Alhady, S. S. (2019). A comprehensive study for robot navigation techniques. *Cogent Engineering*, *6*(1), 1632046.

Guo, Z., Zhang, Y., & Lu, W. (2019). Attention guided graph convolutional networks for relation extraction. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 241–251. https://doi.org/10.18653/v1/P19-1024

Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martı*********************************, M. A., Màrquez, L., Meyers, A., Nivre, J., Padó, S., Štěpánek, J., Straňák, P., Surdeanu, M., Xue, N., & Zhang, Y. (2009). The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, 1–18. https://aclanthology.org/W09-1201

Hakimov, S., Oto, S. A., & Dogdu, E. (2012). Named entity recognition and disambiguation using linked data and graph-based centrality scoring. *Proceedings of the 4th International Workshop on Semantic Web Information Management*. https://doi.org/10.1145/2237867.2237871

Hamza Osman, A., & Barukab, O. (2020). Graph-based text representation and matching: A review of the state of the art and future challenges. *IEEE Access*, *PP*, 1–1. https://doi.org/10.1109/ACCESS.2020.2993191

Han, X., Zhu, H., Yu, P., Wang, Z., Yao, Y., Liu, Z., & Sun, M. (2018). FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 4803–4809. https://doi.org/10.18653/v1/D18-1514

Harnad, S. (1990). The symbol grounding problem. *Physica D Nonlinear Phenomena*, *42*(1-3), 335–346. https://doi.org/10.1016/0167-2789(90)90087-6

Harris, Z. (1954). Distributional structure. *Word*, *10*(2-3), 146–162. https://doi.org/10.1007/978-94-009-8467-7_1

He, H., & Choi, J. (2020). Establishing strong baselines for the new decade: Sequence tagging, syntactic and semantic parsing with bert. *The Thirty-Third International Flairs Conference*.

Heck, L., Hakkani-Tür, D. Z., & Tür, G. (2013). Leveraging knowledge graphs for web-scale unsupervised semantic parsing. *INTERSPEECH*.

Heim, I. (2012). File change semantics and the familiarity theory of definiteness. In R. Bäuerle, C. Schwarze, & A. von Stechow (Eds.), *Meaning, use, and interpretation of language* (pp. 164–189). De Gruyter. https://doi.org/doi:10.1515/9783110852820.164

Heim, I., & Kratzer, A. (1998). *Semantics in generative grammar*. Oxford [u.a.]: Blackwell.

Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., Pennacchiotti, M., Romano, L., & Szpakowicz, S. (2010). SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. *Proceedings of the 5th International Workshop on Semantic Evaluation*, 33–38. https://aclanthology.org/S10-1006

Hendrix, G. G., Sacerdoti, E. D., Sagalowicz, D., & Slocum, J. (1978). Developing a natural language interface to complex data. *ACM Trans. Database Syst.*, *3*(2), 105–147. https://doi.org/10.1145/320251.320253

Hernández, D., Hogan, A., & Krötzsch, M. (2015). Reifying rdf: What works well with wikidata? *1457*, 32–47.

Hershcovich, D., & Donatelli, L. (2021). It's the meaning that counts: The state of the art in nlp and semantics. *KI - Künstliche Intelligenz*, *35*. https://doi.org/10.1007/s13218-021-00726-6

Hitzler, P., Bianchi, F., Ebrahimi, M., & Sarker, M. K. (2019). Neural-symbolic integration and the semantic web a position paper.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*, 1735–80. https://doi.org/10.1162/neco.1997.9.8.1735

Hoffart, J., Suchanek, F., Berberich, K., Lewis-Kelham, E., de Melo, G., & Weikum, G. (2011). Yago2: Exploring and querying world knowledge in time, space, context, and many languages. *Proceedings of the 20th International Conference Companion on World Wide Web, WWW 2011*, 229–232. https://doi.org/10.1145/1963192.1963296

Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., de Melo, G., Gutiérrez, C., Kirrane, S., Labra Gayo, J. E., Navigli, R., Neumaier, S., Ngonga Ngomo, A.-C., Polleres, A., Rashid, S. M., Rula, A., Schmelzeisen, L., Sequeda, J. F., Staab, S., & Zimmermann, A. (2021). *Knowledge Graphs*. Morgan & Claypool. https://doi.org/10.2200/S01125ED1V01Y202109DSK022

Holland, P. W. (1986). Statistics and causal inference. *Journal of the American Statistical Association*, *81*(396), 945–960. Retrieved October 26, 2022, from http://www.jstor.org/stable/2289064

Holm, J., Thomas, G., Hendler, J., & Musialek, C. (2012). Us government linked open data: Semantic.data.gov. *IEEE Intelligent Systems*, *27*(03), 25–31. https://doi.org/10.1109/MIS.2012.27

Horn, F. (2017). Context encoders as a simple but powerful extension of word2vec.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, *24*(6), 417–441. https://doi.org/10.1037/h0071325

Hu, J., Qian, S., Fang, Q., & Xu, C. (2019). Hierarchical graph semantic pooling network for multi-modal community question answer matching. *Proceedings of the 27th ACM International Conference on Multimedia*, 1157–1165. https://doi.org/10.1145/3343031.3350966

Hu, J., Fang, Q., Qian, S., & Xu, C. (2020). Multi-modal attentive graph pooling model for community question answer matching. *Proceedings of the 28th ACM International Conference on Multimedia*, 3505–3513. https://doi.org/10.1145/3394171.3413711

Hu, S., Zou, L., & Zhang, X. (2018). A state-transition framework to answer complex questions over knowledge base. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2098–2108. https://doi.org/10.18653/v1/D18-1234

Huang, Z., Silva, A., & Singh, A. (2021). A broader picture of random-walk based graph embedding. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 685–695. https://doi.org/10.1145/3447548.3467300

Huesmann, L. R. (1974). Computer models of thought and language. *The American Journal of Psychology*, *87*(4), 751–754. http://www.jstor.org/stable/1421990

Hutchens, J. L. (1996). How to pass the turing test by cheating. *School of Electrical, Electronic and Computer Engineering research report TR97-05. Perth: University of Western Australia*.

Icarte, R. T., Baier, J. A., Ruz, C., & Soto, A. (2017). How a general-purpose commonsense ontology can improve performance of learning-based image retrieval. *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 1283–1289.

Iida, R., Inui, K., & Matsumoto, Y. (2006). Exploiting syntactic patterns as clues in zero-anaphora resolution. *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 625–632. https://doi.org/10.3115/1220175.1220254

Inui, N., Koiso, T., Nakamura, J., & Kotani, Y. (2003). Fully corpus-based natural language dialogue system.

Jackendoff, R. (1972). *Semantic interpretation in generative grammar*. MIT Press.

Jackendoff, R. (1990). *Semantic structures*. MIT Press.

Jacquet, B., Jamet, F., & Baratgin, J. (2021). On the pragmatics of the turing test. *2021 International Conference on Information and Digital Technologies (IDT)*, 123–130. https://doi.org/10.1109/IDT52577.2021.9497570

Jawahar, G., Sagot, B., & Seddah, D. (2019). What does BERT learn about the structure of language? In A. Korhonen, D. R. Traum, & L. Màrquez (Eds.), *Proceedings of the 57th conference of the association for computational linguistics, ACL 2019, florence, italy, july 28- august 2, 2019, volume 1: Long papers* (pp. 3651–3657). Association for Computational Linguistics. https://doi.org/10.18653/v1/p19-1356

Ji, S., Pan, S., Cambria, E., Marttinen, P., & Philip, S. Y. (2021). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(2), 494–514.

Jia, R., & Liang, P. (2016). Data recombination for neural semantic parsing. *CoRR*, *abs/1606.03622*. http://arxiv.org/abs/1606.03622

Jia, R., & Liang, P. (2017). Adversarial examples for evaluating reading comprehension systems. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2021–2031. https://doi.org/10.18653/v1/D17-1215

Johnson, T. (1984). Natural language computing: The commercial applications. *The Knowledge Engineering Review*, *1*(3), 11–23. https://doi.org/10.1017/S0269888900000588

Jones, K. S. (1964). *Synonymy and semantic classification*. Cambridge: Eng., Cambridge Language Research Unit.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, *28*, 11–21.

Jones, K. S. (2004). *Language modelling's generative model: Is it rational* (tech. rep.).

Joshi, P., Santy, S., Budhiraja, A., Bali, K., & Choudhury, M. (2020). The state and fate of linguistic diversity and inclusion in the NLP world. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6282–6293. https://doi.org/10.18653/v1/2020.acl-main.560

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2017). Bag of tricks for efficient text classification. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 427–431.

Ju, R., Zhou, P., Li, C. H., & Liu, L. (2015). An efficient method for document categorization based on word2vec and latent semantic analysis. *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, 2276–2283. https://doi.org/10.1109/CIT/IUCC/DASC/PICOM.2015.336

Kamath, A., & Das, R. (2018). A survey on semantic parsing. *CoRR*, *abs/1812.00978*. http://arxiv.org/abs/1812.00978

Kamp, H. (1981). A theory of truth and semantic representation, 277-322, jag groenendijk, tmv janssen and mbj stokhof, eds. In J. Groenendijk (Ed.), *Formal methods in the study of language*. U of Amsterdam.

Kanerva, P., Kristofersson, J., & Holst, A. (2000). Random indexing of text samples for latent semantic analysis. In L. Gleitman & A. Josh (Eds.), *Proceedings of the 22nd annual conference of the cognitive science society*. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.4.6523&rep=rep1&type=pdf

Kanerva, P. (1988). *Sparse distributed memory*. MIT Press.

Kate, R. J., & Wong, Y. W. (2010). Semantic parsing. the task, the state of the art and the future. *Tutorial abstracts of the 20th Meeting of the Association for Computational Linguistics*, 6.

Kim, S., Park, H., & Lee, J. (2020). Word2vec-based latent semantic analysis (w2v-lsa) for topic modeling: A study on blockchain technology trend analysis. *Expert Systems with Applications*, *152*, 113401.

Kipf, T. N., & Welling, M. (2016). Semi-Supervised Classification with Graph Convolutional Networks. *arXiv e-prints*, Article arXiv:1609.02907, arXiv:1609.02907.

Kölbel, M. (2008). Truth in semantics. *Midwest studies in philosophy*, *32*, 242–257.

Kollar, T., & Roy, N. (2009). Utilizing object-object and object-scene context when planning to find things. *2009 IEEE International Conference on Robotics and Automation*, 2168–2173. https://doi.org/10.1109/ROBOT.2009.5152831

Koller, A. (2016). Top-down and bottom-up views on success in semantics. https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxzdGFyc2VtMjAxNnxneDoyNTcyY2VhZmYyYWVkZmFk

Kortenkamp, D., & Weymouth, T. (1994). Topological mapping for mobile robots using a combination of sonar and vision sensing. *Proceedings of the Twelfth AAAI National Conference on Artificial Intelligence*, 979–984.

Krishnamurthy, J., Dasigi, P., & Gardner, M. (2017). Neural semantic parsing with type constraints for semi-structured tables. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1516–1526. https://doi.org/10.18653/v1/D17-1160

Kutuzov, A., & Kuzmenko, E. (2019). To lemmatize or not to lemmatize: How word normalisation affects elmo performance in word sense disambiguation. *arXiv preprint arXiv:1909.03135*.

Kvale, K., Sell, O. A., Hodnebrog, S., & Følstad, A. (2020). Improving conversations: Lessons learnt from manual analysis of chatbot dialogues. *International workshop on chatbot research and design*, 187–200.

Kwiatkowksi, T., Zettlemoyer, L., Goldwater, S., & Steedman, M. (2010). Inducing probabilistic CCG grammars from logical form with higher-order unification. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 1223–1233. https://aclanthology.org/D10-1119

Kwiatkowski, T., Choi, E., Artzi, Y., & Zettlemoyer, L. (2013). Scaling semantic parsers with on-the-fly ontology matching. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1545–1556. https://aclanthology.org/D13-1161

Labov, W. (2001). *Principles of linguistic change* (Vol. 2, Social factors). Blackwell.

Ladyman, J., Lambert, J., & Wiesner, K. (2013). What is a complex system? *European Journal for Philosophy of Science*, *3*(1), 33–67.

Landauer, T. K., & Dumais, S. T. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, *104*, 211–240.

Landgrebe, J., & Smith, B. (2019). There is no general AI: why turing machines cannot pass the turing test. *CoRR*, *abs/1906.05833*. http://arxiv.org/abs/1906.05833

Landgrebe, J., & Smith, B. (2021). An argument for the impossibility of machine intelligence. *CoRR*, *abs/2111.07765*. https://arxiv.org/abs/2111.07765

Landgrebe, J., & Smith, B. (2022). *Why machines will never rule the world: Artificial intelligence without fear*. Taylor & Francis.

Larson, R. K., & Ludlow, P. (1993). Interpreted logical forms. *Synthese*, *95*(3), 305–355. https://doi.org/10.1007/bf01063877

Le, P., & Zuidema, W. (2016). Quantifying the vanishing gradient and long distance dependency problem in recursive neural networks and recursive LSTMs. *Proceedings of the 1st Workshop on Representation Learning for NLP*, 87–93. https://doi.org/10.18653/v1/W16-1610

Lee, T.-W. (1998). Independent component analysis. *Independent component analysis: Theory and applications* (pp. 27–66). Springer US. https://doi.org/10.1007/978-1-4757-2851-4_2

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., & Bizer, C. (2015). Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, *6*, 167–195.

Leitner, E., Rehm, G., & Moreno-Schneider, J. (2020). A dataset of German legal documents for named entity recognition. *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 4478–4485. https://aclanthology.org/2020.lrec-1.551

Lepore, E., & Loewer, B. (1989). You can say that again. *Midwest Studies in Philosophy*, *14*(1), 338–356. https://doi.org/10.1111/j.1475-4975.1989.tb00196.x

Levesque, H. J. (2014). On our best behaviour. *Artificial Intelligence*, *212*, 27–35. https://doi.org/https://doi.org/10.1016/j.artint.2014.03.007

Levin, B. (1993). *English Verb Classes and Alternations A Preliminary Investigation*. University of Chicago Press.

Lewis, D. K. (1970). General semantics. *Synthese*, *22*(1-2), 18–67. https://doi.org/10.1007/bf00413598

Li, J., Luong, T., Jurafsky, D., & Hovy, E. (2015). When are tree structures necessary for deep learning of representations? *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2304–2314. https://doi.org/10.18653/v1/D15-1278

Li, L., Jamieson, K., Rostamizadeh, A., Gonina, E., Ben-tzur, J., Hardt, M., Recht, B., & Talwalkar, A. (2020). A system for massively parallel hyperparameter tuning. In I. Dhillon, D. Papailiopoulos, & V. Sze (Eds.), *Proceedings of machine learning and systems* (pp. 230–246). https://proceedings.mlsys.org/paper/2020/file/f4b9ec30ad9f68f89b29639786cb62ef-Paper.pdf

Liang, B., Su, H., Gui, L., Cambria, E., & Xu, R. (2022). Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks. *Knowledge-Based Systems*, *235*, 107643. https://doi.org/https://doi.org/10.1016/j.knosys.2021.107643

Liang, P. (2016). Learning executable semantic parsers for natural language understanding. *CoRR*, *abs/1603.06677*. http://arxiv.org/abs/1603.06677

Liang, P., Jordan, M. I., & Klein, D. (2011). Learning dependency-based compositional semantics.

Liang, P., Jordan, M. I., & Klein, D. (2013). Learning dependency-based compositional semantics. *Computational Linguistics*, *39*(2), 389–446.

Lin, J. (2007). An exploration of the principles underlying redundancy-based factoid question answering. *ACM Trans. Inf. Syst.*, *25*(2), 6–es. https://doi.org/10.1145/1229179.1229180

Lin, Y., Liu, Z., Sun, M., Liu, Y., & Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2181–2187.

Ling, W., Dyer, C., Black, A. W., & Trancoso, I. (2015). Two/too simple adaptations of word2vec for syntax problems. *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1299–1304.

Linzen, T. (2020). How can we accelerate progress towards human-like linguistic generalization? In D. Jurafsky, J. Chai, N. Schluter, & J. R. Tetreault (Eds.), *Proceedings of the 58th annual meeting of the association for computational linguistics, ACL 2020, online, july 5-10, 2020* (pp. 5210–5217). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.465

Liu, H. (2017). Sentiment analysis of citations using word2vec.

Liu, Q., Kusner, M. J., & Blunsom, P. (2020). A survey on contextual embeddings.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach.

Liu, Z., Huang, W., Zheng, Y., & Sun, M. (2010). Automatic keyphrase extraction via topic decomposition. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, MIT Stata Center, Massachusetts,* *USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, 366–376. https://aclanthology.org/D10-1036/

Long, T., Lowe, R., Cheung, J. C. K., & Precup, D. (2016). Leveraging lexical resources for learning entity embeddings in multi-relational data.

Lovász, L. (1993). Random walks on graphs. *Combinatorics, Paul erdos is eighty*, *2*(1-46), 4.

Luo, K., Lin, F., Luo, X., & Zhu, K. (2018). Knowledge base question answering via encoding of complex query graphs. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2185–2194. https://doi.org/10.18653/v1/D18-1242

Luz, F. F. (2019). *Deep neural semantic parsing: Translating from natural language into sparql* (Doctoral dissertation). Universidade de São Paulo.

Luz, F. F., & Finger, M. (2018). Semantic parsing natural language into sparql: Improving target language representation with neural attention. *arXiv preprint arXiv:1803.04329*.

Luz de Araujo, P. H., de Campos, T. E., de Oliveira, R. R. R., Stauffer, M., Couto, S., & Bermejo, P. (2018). LeNER-Br: A dataset for named entity recognition in Brazilian legal text. *International Conference on the Computational Processing of Portuguese (PROPOR)*, 313–323. https://doi.org/10.1007/978-3-319-99722-3_32

Lv, X., Cao, Y., Hou, L., Li, J., Liu, Z., Zhang, Y., & Dai, Z. (2021). Is multi-hop reasoning really explainable? towards benchmarking reasoning interpretability. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 8899–8911. https://doi.org/10.18653/v1/2021.emnlp-main.700

M K, V., & K, K. (2016). A survey on similarity measures in text mining. *Machine Learning and Applications: An International Journal*, *3*, 19–28. https://doi.org/10.5121/mlaij.2016.3103

Ma, K., Ilievski, F., Francis, J., Bisk, Y., Nyberg, E., & Oltramari, A. (2020). Knowledge-driven data construction for zero-shot evaluation in commonsense question answering.

MacDonald, M. C., Pearlmutter, N. J., & Seidenberg, M. S. (1994). The lexical nature of syntactic ambiguity resolution [corrected]. *Psychological review*, *101 4*, 676–703.

Maienborn, C., von Heusinger, K., & Portner, P. (Eds.). (2019). *Semantics - theories*. De Gruyter Mouton. https://doi.org/doi:10.1515/9783110589245

Mallick, C., Das, A., Dutta, M., Das, A., & Sarkar, A. (2018). Graph-based text summarization using modified textrank. *Advances in Intelligent Systems and Computing*. https://doi.org/10.1007/978-981-13-0514-6_14

Manola, F., & Miller, E. (2004). Rdf primer. w3c recommendation, w3c, 2004. https://www.w3.org/TR/rdf-primer/

Marcheggiani, D., & Titov, I. (2017). Encoding sentences with graph convolutional networks for semantic role labeling. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1506–1515. https://doi.org/10.18653/v1/D17-1159

Marcheggiani, D., Bastings, J., & Titov, I. (2018). Exploiting semantics in neural machine translation with graph convolutional networks. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 486–492. https://doi.org/10.18653/v1/N18-2078

Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. *Comput. Linguist.*, *19*(2), 313–330.

Maturana, R., Alvarado-Cortes, E., López Sola, S., Martínez-Losa, M., & Hermoso-González, P. (2018). La rioja turismo: The construction and exploitation of a queryable tourism knowledge graph: Icwe 2018 international workshops, matwep, enwot, kd-web, weod, tourismkg, cáceres, spain, june 5, 2018, revised selected papers. https://doi.org/10.1007/978-3-030-03056-8_20

McCarthy, J. (1968). The advice taker. In M. L. Minsky (Ed.), *Semantic information processing*. MIT Press.

McCoy, T., Pavlick, E., & Linzen, T. (2019). Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3428–3448. https://doi.org/10.18653/v1/P19-1334

McCrae, J., Spohr, D., & Cimiano, P. (2011). Linking lexical resources and ontologies on the semantic web with lemon. *Extended Semantic Web Conference*, 245–259.

McCrae, J. P., Bosque-Gil, J., Gracia, J., Buitelaar, P., & Cimiano, P. (2017). The ontolex-lemon model: Development and applications. *Proceedings of eLex 2017 conference*, 19–21.

Melis, G., Dyer, C., & Blunsom, P. (2017). On the state of the art of evaluation in neural language models.

Meyer, I. (2001). Extracting knowledge-rich contexts for terminography. *Recent advances in computational terminology*, *2*, 279.

Mihalcea, R. (2004). Graph-based ranking algorithms for sentence extraction, applied to text summarization. *ACL*.

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into text. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 404–411. https://aclanthology.org/W04-3252

Mihalcea, R., & Radev, D. (2011). *Graph-based natural language processing and information retrieval*. Cambridge University Press. https://doi.org/10.1017/CBO9780511976247

Mikheev, A. (1997a). Automatic rule induction for unknown-word guessing. *Comput. Linguistics*, *23*, 405–423.

Mikheev, A. (1997b). Automatic rule induction for unknown-word guessing. *Comput. Linguist.*, *23*(3), 405–423.

Mikolov, T., Yih, W.-t., & Zweig, G. (2013). Linguistic regularities in continuous space word representations. *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 746–751. https://aclanthology.org/N13-1090

Mikolov, T., Yih, W., & Zweig, G. (2013). Linguistic regularities in continuous space word representations. In L. Vanderwende, H. D. III, & K. Kirchhoff (Eds.), *Human language technologies: Conference of the north american chapter of the association of computational linguistics, proceedings, june 9-14, 2013, westin peachtree plaza hotel, atlanta, georgia, USA* (pp. 746–751). The Association for Computational Linguistics. https://aclanthology.org/N13-1090/

Miller, G. A. (1995). Wordnet: A lexical database for english. *Communications of the ACM*, *38*(11), 39–41.

Mills, M., & Bourbakis, N. (2014). Graph-based methods for natural language processing and understanding—a survey and analysis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *44*, 59–71. https://doi.org/10.1109/TSMCC.2012.2227472

Minsky, M. (1974). *A framework for representing knowledge* (tech. rep.). USA, Massachusetts Institute of Technology.

Mintz, M., Bills, S., Snow, R., & Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. *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*, 1003–1011. https://aclanthology.org/P09-1113

Mitkov, R. (2001). Outstanding issues in anaphora resolution, 110–125. https://doi.org/10.1007/3-540-44686-9_11

Miwa, M., & Bansal, M. (2016). End-to-end relation extraction using lstms on sequences and tree structures. *arXiv preprint arXiv:1601.00770*.

Mochales, R., & Moens, M.-F. (2011). Argumentation mining. *Artificial Intelligence and Law*, *19*, 1–22. https://doi.org/10.1007/s10506-010-9104-x

Modi, A., Titov, I., & Klementiev, A. (2012). Unsupervised induction of frame-semantic representations. *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*, 1–7.

Montague, R. (1970a). Universal grammar. *Theoria*, *36*(3), 373–398. https://doi.org/10.1111/j.1755-2567.1970.tb00434.x

Montague, R. (1970b). Universal grammar. *Theoria*, *36*(3), 373–398. https://doi.org/10.1111/j.1755-2567.1970.tb00434.x

Montague, R. (1974). The proper treatment of quantification in ordinary english. In R. H. Thomason (Ed.), *Formal philosophy*. Yale University Press.

Moravec, H. (1988). Sensor fusion in certainty grids for mobile robots. *AI Mag.*, *9*(2), 61–74.

Mukherjee, S., & Bhattacharyya, P. (2012). Feature specific sentiment analysis for product reviews. In A. Gelbukh (Ed.), *Computational linguistics and intelligent text processing* (pp. 475–487). Springer Berlin Heidelberg.

Musolino, J. (2006). Structure and meaning in the acquisition of scope. https://doi.org/10.1007/1-4020-4485-2_6

Naseem, U., Razzak, I., Khan, S. K., & Prasad, M. (2021). A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models. *ACM Trans. Asian Low Resour. Lang. Inf. Process.*, *20*(5), 74:1–74:35. https://doi.org/10.1145/3434237

Nastase, V., Mihalcea, R., & Radev, D. R. (2015). A survey of graphs in natural language processing. *Natural Language Engineering*, *21*(5), 665–698.

Nguyen, M.-T., Phan, V.-A., Linh, L. T., Son, N. H., Dung, L. T., Hirano, M., & Hotta, H. (2020). Transfer learning for information extraction with limited data. *Computational Linguistics: 16th International Conference of the Pacific Association for Computational Linguistics, PACLING 2019, Hanoi, Vietnam, October 11–13, 2019, Revised Selected Papers 16*, 469–482.

Nguyen, V., Bodenreider, O., & Sheth, A. (2014). Don't like rdf reification? making statements about statements using singleton property. *Proceedings of the 23rd International Conference on World Wide Web*, 759–770. https://doi.org/10.1145/2566486.2567973

Nickel, M., Tresp, V., & Kriegel, H.-P. (2011). A three-way model for collective learning on multi-relational data. *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 809–816.

Niven, T., & Kao, H.-Y. (2019). Probing neural network comprehension of natural language arguments. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4658–4664. https://doi.org/10.18653/v1/P19-1459

Niwa, Y., & Nitta, Y. (2003). Co-occurrence vectors from corpora vs. distance vectors from dictionaries. https://doi.org/10.3115/991886.991938

Nolano, G., Elahi, M., di Buono, M., Ell, B., & Cimiano, P. (2021). An italian question answering system based on grammars automatically generated from ontology lexica. *Proceedings of the Eighth Italian Conference on Computational Linguistics (CLIC-it 2021)*.

Nolano, G., di Buono, M. P., & Monti, J. (2022a). From monolingual multiword expression discovery to multilingual concept enrichment: An ontology-based approach. *The 5th Workshop on Multi-word units in Machine Translation and Translation Technology (MUMTTT 2022)*.

Nolano, G., di Buono, M. P., & Monti, J. (2022b). *Unior nlp@ cofe datathon 2022 task 1-keyword extraction from proposals*. COFE Datathon. https://prod-cofe-platform.s3.eu-central-1.amazonaws.com/9nru2951mlj41gebe1gzt9uf9xkw?response-content-disposition=inlin e%3B%20filename%3D%22COFE-Datathon-UNIOR_NLP-Challenge_1.pdf%22%3B%20filename%2A%3DUTF-8%27%27COFE-Datathon-UNIOR_NLP-Challenge_1.pdf&response-content-type=application%2Fpdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIA3LJJXGZPDFYVOW5V%2F20221104%2Feu-central-1%2Fs3%2Faws4_request&X-Amz-Date=20221104T091854Z&X-Amz-Expires=300&X-Amz-SignedHeaders=host&X-Amz-Signature=3eca5795e4b1688a6887553865fcbd2bf2ea9a9210f26daa54880839167af58f

Nolano, G., di Buono, M. P., & Monti, J. (2022c). *Unior nlp@ cofe datathon 2022 task 2-idea clustering*. COFE Datathon. https://prod-cofe-platform.s3.eu-central-1.amazonaws.com/0ki4xwhis14wf5w8cnhl0t4muo48?response-content-disposition=inline%3B%20filename%3D%22COFE-Datathon-UNIOR_NLP-Challenge_2.pdf%22%3B%20filename%2A%3DUTF-8%27%27COFE-Datathon-UNIOR_NLP-Challenge_2.pdf&response-content-type=application%2Fpdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIA3LJJXGZPDFYVOW5V%2F20221116%2Feu-central-1%2Fs3%2Faws4_request&X-Amz-Date=20221116T092326Z&X-Amz-Expires=300&X-Amz-SignedHeaders=host&X-Amz-Signature=daae766d89c81e3d9f75abb9e18fefe6e9bd4108c4ef16ee602d77c0ecad57dd

Nüchter, A., & Hertzberg, J. (2008). Towards semantic maps for mobile robots [Semantic Knowledge in Robotics]. *Robotics and Autonomous Systems*, *56*(11), 915–926. https://doi.org/https://doi.org/10.1016/j.robot.2008.08.001

Oren, I., Herzig, J., Gupta, N., Gardner, M., & Berant, J. (2020). Improving compositional generalization in semantic parsing. *arXiv preprint arXiv:2010.05647*.

Padó, S., & Lapata, M. (2005). Cross-linguistic projection of role-semantic information. *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 859–866.

Palmer, M., Gildea, D., & Kingsbury, P. (2005a). The proposition bank: An annotated corpus of semantic roles. *Comput. Linguist.*, *31*(1), 71–106. https://doi.org/10.1162/0891201053630264

Palmer, M., Gildea, D., & Kingsbury, P. (2005b). The proposition bank: An annotated corpus of semantic roles. *Comput. Linguist.*, *31*(1), 71–106. https://doi.org/10.1162/0891201053630264

Palmero Aprosio, A., & Moretti, G. (2018). Tint 2.0: An all-inclusive suite for nlp in italian. *10*, 12.

Papernot, N., McDaniel, P., Swami, A., & Harang, R. (2016). Crafting adversarial input sequences for recurrent neural networks. *MILCOM 2016-2016 IEEE Military Communications Conference*, 49–54.

Partee, B. H. (2010). Formal semantics: Origins, issues, early impact. *The Baltic International Yearbook of Cognition, Logic and Communication*, *6*(1). https://doi.org/10.4148/biyclc.v6i0.1580

Partee, B. H. (1981). Montague grammar, mental representations, and reality. *Philosophy and Grammar: Papers on the Occasion of the Quincentennial of Uppsala University*, 59–78.

Passos, A., Kumar, V., & McCallum, A. (2014). Lexicon infused phrase embeddings for named entity resolution. *CoRR*, *abs/1404.5367*. http://arxiv.org/abs/1404.5367

Pasupat, P., & Liang, P. (2015). Compositional semantic parsing on semi-structured tables.

Pavlik, J. V. (2023). Collaborating with chatgpt: Considering the implications of generative artificial intelligence for journalism and media education. *Journalism & Mass Communication Educator*, 10776958221149577.

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *2*(11), 559–572. https://doi.org/10.1080/14786440109462720

Pelgrin, O., Galárraga, L., & Hose, K. (2021). Towards fully-fledged archiving for rdf datasets. *Semantic Web*, (Preprint), 1–24.

Peñas, A., Verdejo, F., & Gonzalo, J. (2001). Corpus-based terminology extraction applied to information access.

Peng, N., Poon, H., Quirk, C., Toutanova, K., & Yih, W.-t. (2017). Cross-sentence n-ary relation extraction with graph LSTMs. *Transactions of the Association for Computational Linguistics*, *5*, 101–115. https://doi.org/10.1162/tacl_a_00049

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *EMNLP*, *14*, 1532–1543.

Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). Deepwalk: Online learning of social representations. *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 701–710. https://doi.org/10.1145/2623330.2623732

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2227–2237. https://doi.org/10.18653/v1/N18-1202

Peters, M., Neumann, M., Logan, R. L. I., Schwartz, R., Joshi, V., Singh, S., & Smith, N. A. (2019). Knowledge enhanced contextual word representations.

Petrolito, R., & Dell'Orletta, F. (2018). Word embeddings in sentiment analysis. *CLiC-it*.

Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A. H., & Riedel, S. (2019). Language models as knowledge bases?

Phillips, A. V. (1960). *A question-answering routine* (tech. rep.). USA, Massachusetts Institute of Technology.

Phillips, C., Kazanina, N., & Abada, S. H. (2005). Erp effects of the processing of syntactic long-distance dependencies. *Cognitive Brain Research*, *22*(3), 407–428.

Piñango, M. M., Finn, E., Lacadie, C., & Constable, R. T. (2016). The localization of long-distance dependency components: Integrating the focal-lesion and neuroimaging record. *Frontiers in psychology*, *7*, 1434.

Pinkal, M. (1995). *Logic and lexicon the semantics of the indefinite*. Springer.

Pirozelli, P., & Câmara, I. (2022). Natural language at a crossroads: Formal and probabilistic approaches in philosophy and computer science. *Manuscrito*, *45*(2), 50–81. https://doi.org/10.1590/0100-6045.2022.v45n2.pi

Pörner, N., Waltinger, U., & Schütze, H. (2019). BERT is not a knowledge base (yet): Factual knowledge vs. name-based reasoning in unsupervised QA. *CoRR*, *abs/1911.03681*. http://arxiv.org/abs/1911.03681

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., & Huang, X. (2020a). Pre-trained models for natural language processing: A survey. *CoRR*, *abs/2003.08271*. https://arxiv.org/abs/2003.08271

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., & Huang, X. (2020b). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, *63*, 1872–1897. https://doi.org/10.1007/s11431-020-1647-3

Quillian, M. R. et al. (1962). A revised design for an understanding machine. *Mech. Transl. Comput. Linguistics*, *7*(1), 17–29.

Quillian, R. (1968). Semantic memory. *Semantic information processing* (pp. 216–270). MIT Press.

Quirk, C., & Poon, H. (2017). Distant supervision for relation extraction beyond the sentence boundary, 1171–1182. https://doi.org/10.18653/v1/E17-1110

Rabbani, K., Lissandrini, M., & Hose, K. (2021). Optimizing sparql eries using shape statistics.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2383–2392. https://doi.org/10.18653/v1/D16-1264

Rajpurkar, P., Jia, R., & Liang, P. (2018). Know what you don't know: Unanswerable questions for SQuAD. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 784–789. https://doi.org/10.18653/v1/P18-2124

Ramshaw, L. A., & Marcus, M. P. (1999). Text chunking using transformation-based learning. *Natural language processing using very large corpora* (pp. 157–176). Springer.

Thompson, C. A., Califf, M. E., & Mooney, R. J. (1999). Active learning for natural language parsing and information extraction. *Proceedings of the Sixteenth International Conference on Machine Learning (ICML-99)*, 406–414. http://www.cs.utexas.edu/users/ai-lab?thompson:ml99

Rashkin, H., Sap, M., Allaway, E., Smith, N. A., & Choi, Y. (2018). Event2Mind: Commonsense inference on events, intents, and reactions. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 463–473. https://doi.org/10.18653/v1/P18-1043

Ribeiro, E., Teixeira, A. S., Ribeiro, R., & Martins de Matos, D. (2020). Semantic frame induction through the detection of communities of verbs and their arguments. *Applied Network Science*, *5*(1), 1–32.

Ristoski, P., & Paulheim, H. (2016). Rdf2vec: Rdf graph embeddings for data mining. *International Semantic Web Conference*, 498–514.

Ristoski, P., Rosati, J., Di Noia, T., De Leone, R., & Paulheim, H. (2018). Rdf2vec: Rdf graph embeddings and their applications [in press]. *Semantic Web Journal*. http://sisinflab.poliba.it/Publications/2018/RRDDP18b

Roeper, T. (2010). The acquisition of recursion: How formalism articulates the child's path. *Biolinguistics*, *5*. https://doi.org/10.5964/bioling.8831

Rongali, S., Soldaini, L., Monti, E., & Hamza, W. (2020). Don't parse, generate! a sequence to sequence architecture for task-oriented semantic parsing. *Proceedings of The Web Conference 2020*, 2962–2968.

Rospocher, M., Corcoglioniti, F., & Palmero Aprosio, A. (2019). Premon: Lodifing linguistic predicate models. *Language Resources and Evaluation*, *53*(3), 499–524.

Roy, A., & Pan, S. (2021). Incorporating extra knowledge to enhance word embedding. *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 4929–4935.

Rubenstein, H., & Goodenough, J. B. (1965). Contextual correlates of synonymy. *Commun. ACM*, *8*(10), 627–633. https://doi.org/10.1145/365628.365657

Ruzzetti, E. S., Ranaldi, L., Mastromattei, M., Fallucchi, F., Scarpato, N., & Zanzotto, F. M. (2022). Lacking the embedding of a word? look it up into a traditional dictionary. *Findings of the Association for Computational Linguistics: ACL 2022*, 2651–2662. https://doi.org/10.18653/v1/2022.findings-acl.208

Sacerdoti, E. D. (1977). Language access to distributed data with error recovery. *IJCAI*.

Safavi, T., & Koutra, D. (2021). Relational world knowledge representation in contextual language models: A review.

Sahlgren, M. (2005). An introduction to random indexing. *Methods and applications of semantic indexing workshop at the 7th international conference on terminology and knowledge engineering*.

Sahu, S. K., Christopoulou, F., Miwa, M., & Ananiadou, S. (2019). Inter-sentence relation extraction with document-level graph convolutional neural network. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4309–4316. https://doi.org/10.18653/v1/P19-1423

Saleem, M., Potocki, A., Soru, T., Hartig, O., & Ngomo, A.-C. N. (2018). Costfed: Cost-based query optimization for sparql endpoint federation. *Procedia Computer Science*, *137*, 163–174.

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

Salton, G., Singhal, A., Mitra, M., & Buckley, C. (1997). Automatic text structuring and summarization. *Inf. Process. Manage.*, *33*(2), 193–207. https://doi.org/10.1016/S0306-4573(96)00062-3

Saygin, A. P., & Çiçekli, I. (2002). Pragmatics in human-computer conversations*. *Journal of Pragmatics*, *34*, 227–258.

Schick, T., & Schütze, H. (2020). Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(05), 8766–8774. https://doi.org/10.1609/aaai.v34i05.6403

Schiffer, S. R. (1987). *Remnants of meaning*. MIT Press.

Schlichtkrull, M., Kipf, T. N., Bloem, P., Berg, R. v. d., Titov, I., & Welling, M. (2018). Modeling relational data with graph convolutional networks. *European semantic web conference*, 593–607.

Schneider, E. W., & Human Resources Research Organization, V., Alexandria. (1973). *Course modularization applied [microform] : The interface system and its implications for sequence control and data analysis / e. w. schneider*. Distributed by ERIC Clearinghouse [Washington, D.C.] https://eric.ed.gov/?id=ED088424

Scontras, G., & Pearl, L. S. (2021). When pragmatics matters more for truth-value judgments: An investigation of quantifier scope ambiguity. *Glossa: a journal of general linguistics*.

Searle, J. R. (1980). Minds, Brains, and Programs. *Behavioral and Brain Sciences*, *3*(03), 417–424. https://doi.org/10.1017/s0140525x00005756

Shah, P., Fiser, M., Faust, A., Kew, J. C., & Hakkani-Tur, D. (2018). Follownet: Robot navigation by following natural language directions with deep reinforcement learning.

Sheikhshab, G., Birol, I., & Sarkar, A. (2018). In-domain context-aware token embeddings improve biomedical named entity recognition. *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, 160–164.

Sheth, A., Perera, S., Wijeratne, S., & Thirunarayan, K. (2017). Knowledge will propel machine understanding of content: Extrapolating from current examples. *Proceedings of the International Conference on Web Intelligence*, 1–9. https://doi.org/10.1145/3106426.3109448

Sheth, A., Gaur, M., Kursuncu, U., & Wickramarachchi, R. (2019). Shades of knowledge-infused learning for enhancing deep learning. *IEEE Internet Computing*, *23*(6), 54–63. https://doi.org/10.1109/MIC.2019.2960071

Shi, T., Malioutov, I., & Irsoy, O. (2020). Semantic role labeling as syntactic dependency parsing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 7551–7571. https://doi.org/10.18653/v1/2020.emnlp-main.610

Sienčnik, S. K. (2015). Adapting word2vec to named entity recognition. *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, 239–243.

Simmons, R. F. (1965). Answering english questions by computer: A survey. *Commun. ACM*, *8*, 53–70.

Simmons, R. F., Klein, S., & McConlogue, K. (1964). Indexing and dependency logic for answering english questions. *American Documentation*, *15*(3), 196–204.

Singhal, A. (2012). *Introducing the knowledge graph: Things, not strings* [2020-11-13]. https://www.blog.google/products/search/introducing-knowledge-graph-things-not/

Soames, S. (2002). *Beyond rigidity: The unfinished semantic agenda of naming and necessity*. Oxford University Press.

Socher, R., Chen, D., Manning, C. D., & Ng, A. Y. (2013). Reasoning with neural tensor networks for knowledge base completion. *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, 926–934.

Song, L., Zhang, Y., Wang, Z., & Gildea, D. (2018). A graph-to-sequence model for amr-to-text generation. *ACL*.

Song, L., Gildea, D., Zhang, Y., Wang, Z., & Su, J. (2019). Semantic neural machine translation using amr. *Transactions of the Association for Computational Linguistics*, *7*, 19–31. https://doi.org/10.1162/tacl_a_00252

Sorokin, D., & Gurevych, I. (2018). Modeling semantics with gated graph neural networks for knowledge base question answering. *Proceedings of the 27th International Conference on Computational Linguistics*, 3306–3317. https://aclanthology.org/C18-1280

Sowa, J. F. (1999). *Knowledge representation: Logical, philosophical and computational foundations*. Brooks/Cole Publishing Co.

Speaks, J. (2021). Theories of Meaning. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy* (Spring 2021). Metaphysics Research Lab, Stanford University.

Speranza, G., di Buono, M. P., Monti, J., & Sangati, F. (2020). From linguistic resources to ontology-aware terminologies: Minding the representation gap. *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 2503–2510. https://aclanthology.org/2020.lrec-1.305

Speranza, G., di Buono, M. P., & Monti, J. (2021). Terms and appositions: What unstructured texts tell us. In M. Bigey, A. Richeton, M. Silberztein, & I. Thomas (Eds.), *Formalizing natural languages: Applications to natural language processing and digital humanities* (pp. 219–230). Springer International Publishing.

Steedman, M. (1996). *A very short introduction to ccg* (tech. rep.).

Steedman, M. (2000). *The syntactic process*. MIT Press.

Sterckx, L., Demeester, T., Deleu, J., & Develder, C. (2015). Topical word importance for fast keyphrase extraction. *Proceedings of the 24th International Conference on World Wide Web*, 121–122. https://doi.org/10.1145/2740908.2742730

Stojnić, U. (2019). Content in a dynamic context. *Noûs*, *53*(2), 394–432. https://doi.org/10.1111/nous.12220

Sukthanker, R., Poria, S., Cambria, E., & Thirunavukarasu, R. (2020). Anaphora and coreference resolution: A review. *Information Fusion*, *59*, 139–162. https://doi.org/https://doi.org/10.1016/j.inffus.2020.01.010

Sun, C., Gong, Y., Wu, Y., Gong, M., Jiang, D., Lan, M., Sun, S., & Duan, N. (2019). Joint type inference on entities and relations via graph convolutional networks. *Proceedings of the*

57th Annual Meeting of the Association for Computational Linguistics, 1361–1370. https://doi.org/10.18653/v1/P19-1131

Sun, K., Yu, D., Yu, D., & Cardie, C. (2019). Improving machine reading comprehension with general reading strategies.

Sun, K., Zhang, R., Mensah, S., Mao, Y., & Liu, X. (2019). Aspect-level sentiment analysis via convolution over dependency tree. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 5679–5688. https://doi.org/10.18653/v1/D19-1569

Sun, Y., Tang, D., Duan, N., Gong, Y., Feng, X., Qin, B., & Jiang, D. (2019). Neural semantic parsing in low-resource settings with back-translation and meta-learning.

Susnjak, T. (2022). Chatgpt: The end of online exam integrity? *arXiv preprint arXiv:2212.09292*.

Šuster, S., & Daelemans, W. (2018). CliCR: A dataset of clinical case reports for machine reading comprehension. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1551–1563. https://doi.org/10.18653/v1/N18-1140

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., & Fergus, R. (2014). Intriguing properties of neural networks. In Y. Bengio & Y. LeCun (Eds.), *2nd international conference on learning representations, ICLR 2014, banff, ab, canada, april 14-16, 2014, conference track proceedings*. http://arxiv.org/abs/1312.6199

Tai, K. S., Socher, R., & Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 1556–1566. https://doi.org/10.3115/v1/P15-1150

Tamari, R., Shani, C., Hope, T., Petruck, M. R. L., Abend, O., & Shahaf, D. (2020). Language (re)modelling: Towards embodied language understanding. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6268–6281. https://doi.org/10.18653/v1/2020.acl-main.559

Tarski, A. (1944a). The semantic conception of truth. In S. Blackburn & K. Simmons (Eds.), *Truth*. Oxford University Press.

Tarski, A. (1944b). The semantic conception of truth: And the foundations of semantics. *Philosophy and phenomenological research*, *4*(3), 341–376.

Temperley, D., & Gildea, D. (2018). Minimizing syntactic dependency lengths: Typological/cognitive universal? *Annual Review of Linguistics*, *4*. https://doi.org/10.1146/annurev-linguistics-011817-045617

Templeton, M., & Burger, J. (1983). Problems in natural-language interface to dbms with examples from eufid. *Proceedings of the First Conference on Applied Natural Language Processing*, 3–16. https://doi.org/10.3115/974194.974197

Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R. T., Kim, N., Durme, B. V., Bowman, S. R., Das, D., & Pavlick, E. (2019). What do you learn from context? probing for sentence structure in contextualized word representations. *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. https://openreview.net/forum?id=SJzSgnRcKX

Tenney, I., Wexler, J., Bastings, J., Bolukbasi, T., Coenen, A., Gehrmann, S., Jiang, E., Pushkarna, M., Radebaugh, C., Reif, E., & Yuan, A. (2020). The language interpretability tool: Extensible, interactive visualizations and analysis for NLP models. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 107–118. https://doi.org/10.18653/v1/2020.emnlp-demos.15

Thanapalasingam, T., van Berkel, L., Bloem, P., & Groth, P. (2022). Relational graph convolutional networks: A closer look. *PeerJ Computer Science*, *8*, e1073.

Thomason, J., Zhang, S., Mooney, R., & Stone, P. (2015). Learning to interpret natural language commands through human-robot dialog. *Proceedings of the 24th International Conference on Artificial Intelligence*, 1923–1929.

Thompson, C. (2003). Acquiring word-meaning mappings for natural language interfaces. *Journal of Artificial Intelligence Research*, *18*, 1–44. https://doi.org/10.1613/jair.1063

Thornton, R. (2008). William snyder, child language: The parametric approach . oxford: Oxford university press, 2007. pp. xii+209. *Journal of Linguistics*, *44*, 545–552.

Thrun, S. (1998). Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, *99*(1), 21–71.

Trichelair, P., Emami, A., Trischler, A., Suleman, K., & Cheung, J. C. K. (2019). How reasonable are common-sense reasoning tasks: A case-study on the Winograd schema challenge and SWAG. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3382–3387. https://doi.org/10.18653/v1/D19-1335

Turian, J., Ratinov, L.-A., & Bengio, Y. (2010). Word representations: A simple and general method for semi-supervised learning. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 384–394. https://aclanthology.org/P10-1040

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, *59*(236), 433–460. http://www.jstor.org/stable/2251299

Usami, Y., Cho, H.-C., Okazaki, N., & Tsujii, J. (2011). Automatic acquisition of huge training data for bio-medical named entity recognition. *Proceedings of BioNLP 2011 Workshop*, 65–73. https://aclanthology.org/W11-0208

Valdez, D., Pickett, A. C., & Goodson, P. (2018). Topic modeling: Latent semantic analysis for the social sciences. *Social Science Quarterly*, *99*(5), 1665–1679.

van Noord, R., Toral, A., & Bos, J. (2020). Character-level representations improve drs-based semantic parsing even in the age of bert.

Vashishth, S., Bhandari, M., Yadav, P., Rai, P., Bhattacharyya, C., & Talukdar, P. (2019). Incorporating syntactic and semantic information in word embeddings using graph convolutional networks. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3308–3318. https://doi.org/10.18653/v1/P19-1320

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *CoRR*, *abs/1706.03762*. http://arxiv.org/abs/1706.03762

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2017). Graph attention networks. https://doi.org/10.48550/ARXIV.1710.10903

Venhuizen, N. J., Hendriks, P., Crocker, M. W., & Brouwer, H. (2022). Distributional formal semantics [Special Issue: Selected Papers from WoLLIC 2019, the 26th Workshop on

Logic, Language, Information and Computation]. *Information and Computation*, *287*, 104763. https://doi.org/https://doi.org/10.1016/j.ic.2021.104763

Vrandečić, D., & Krötzsch, M. (2014). Wikidata: A free collaborative knowledgebase. *Commun. ACM*, *57*(10), 78–85. https://doi.org/10.1145/2629489

Waltl, B., Bonczek, G., & Matthes, F. (2018). Rule-based information extraction: Advantages, limitations, and perspectives. *Jusletter IT (02 2018)*.

Wan, X., & Xiao, J. (2008). Single document keyphrase extraction using neighborhood knowledge. In D. Fox & C. P. Gomes (Eds.), *Proceedings of the twenty-third AAAI conference on artificial intelligence, AAAI 2008, chicago, illinois, usa, july 13-17, 2008* (pp. 855–860). AAAI Press. http://www.aaai.org/Library/AAAI/2008/aaai08-136.php

Wang, C., Yan, M., Yi, C., & Sha, Y. (2019). Capturing semantic and syntactic information for link prediction in knowledge graphs. *The Semantic Web – ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part I*, 664–679. https://doi.org/10.1007/978-3-030-30793-6_38

Wang, H.-Y., & Ma, W.-Y. (2017). Integrating semantic knowledge into lexical embeddings based on information content measurement. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 509–515. https://aclanthology.org/E17-2082

Wang, J., Yu, L.-C., Lai, K. R., & Zhang, X. (2016). Dimensional sentiment analysis using a regional cnn-lstm model. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 225–230.

Wang, J., Sun, C., Li, S., Wang, J., Si, L., Zhang, M., Liu, X., & Zhou, G. (2019). Human-like decision making: Document-level aspect sentiment classification via hierarchical reinforcement learning. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 5581–5590. https://doi.org/10.18653/v1/D19-1560

Wang, K., Shen, W., Yang, Y., Quan, X., & Wang, R. (2020). Relational graph attention network for aspect-based sentiment analysis. https://doi.org/10.48550/ARXIV.2004.12362

Wang, M., Qiu, L., & Wang, X. (2021). A survey on knowledge graph embeddings for link prediction. *Symmetry*, *13*, 485.

Wang, Q., Mao, Z., Wang, B., & Guo, L. (2017). Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, *29*(12), 2724–2743. https://doi.org/10.1109/TKDE.2017.2754499

Wang, R., Tang, D., Duan, N., Wei, Z., Huang, X., Ji, J., Cao, G., Jiang, D., & Zhou, M. (2021). K-adapter: Infusing knowledge into pre-trained models with adapters. *FINDINGS*.

Wang, X., Jiang, Y., Bach, N., Wang, T., Huang, Z., Huang, F., & Tu, K. (2021). Automated concatenation of embeddings for structured prediction.

Wang, Z., Zhang, J., Feng, J., & Chen, Z. (2014). Knowledge graph embedding by translating on hyperplanes. *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 1112–1119.

Warstadt, A., Cao, Y., Grosu, I., Peng, W., Blix, H., Nie, Y., Alsop, A., Bordia, S., Liu, H., Parrish, A., Wang, S., Phang, J., Mohananey, A., Htut, P. M., Jeretic, P., & Bowman, S. R. (2019). Investigating bert's knowledge of language: Five analysis methods with npis. In K. Inui, J. Jiang, V. Ng, & X. Wan (Eds.), *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing, EMNLP-IJCNLP 2019, hong kong, china, november 3-7, 2019* (pp. 2877–2887). Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1286

Weizenbaum, J. (1966). Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, *9*(1), 36–45. https://doi.org/10.1145/365153.365168

Weizenbaum, J. (1976). *Computer power and human reason*. Freeman.

Westhelle, M., Bencke, L., & Moreira, V. P. (2022). Impact of morphological segmentation on pre-trained language models. *Intelligent Systems: 11th Brazilian Conference, BRACIS 2022, Campinas, Brazil, November 28–December 1, 2022, Proceedings, Part II*, 402–416.

Wheeler, M. R. (1999). Semantics in aristotle's organon. *Journal of the History of Philosophy*, *37*(2), 191–226.

Wintner, S. (2002). Formal language theory for natural language processing. *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, 71–76.

Woods, W. (1973). Progress in natural language understanding: An application to lunar geology. *Proceedings of the June 4-8, 1973, National Computer Conference and Exposition*, 441–450. https://doi.org/10.1145/1499586.1499695

Woods, W., Kaplan, R., & Webber, B. (1972). The lunar science natural language information system: Final report.

Wu, L., Chen, Y., Kai, S., Guo, X., Gao, H., Li, S., Pei, J., & Long, B. (2021). Graph neural networks for natural language processing: A survey.

Wu, Q., Xing, C., Li, Y., Ke, G., He, D., & Liu, T.-Y. (2020). Taking notes on the fly helps language pre-training. *International Conference on Learning Representations*.

Wu, S., & He, Y. (2019). Enriching pre-trained language model with entity information for relation classification. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2361–2364. https://doi.org/10.1145/3357384.3358119

Wu, S., & Dredze, M. (2020). Are all languages created equal in multilingual BERT? *Proceedings of the 5th Workshop on Representation Learning for NLP*, 120–130. https://doi.org/10.18653/v1/2020.repl4nlp-1.16

Wu, Y., Jiang, M., Lei, J., & Xu, H. (2015). Named entity recognition in chinese clinical text using deep neural network. *Studies in health technology and informatics*, *216*, 624–8.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., … Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, *abs/1609.08144*.

Xu, K., Feng, Y., Huang, S., & Zhao, D. (2015). Semantic relation classification via convolutional neural networks with simple negative sampling. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 536–540. https://doi.org/10.18653/v1/D15-1062

Xu, W., Hoffmann, R., Zhao, L., & Grishman, R. (2013). Filling knowledge base gaps for distant supervision of relation extraction. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 665–670. https://aclanthology.org/P13-2117

Yanaka, H., Mineshima, K., Bekki, D., Inui, K., Sekine, S., Abzianidze, L., & Bos, J. (2019). Can neural networks understand monotonicity reasoning? *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 31–40. https://doi.org/10.18653/v1/W19-4804

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2020). Xlnet: Generalized autoregressive pretraining for language understanding.

Yang, Z., Blunsom, P., Dyer, C., & Ling, W. (2017). Reference-aware language models.

Yao, L., Mao, C., & Luo, Y. (2019). Graph convolutional networks for text classification. *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*. https://doi.org/10.1609/aaai.v33i01.33017370

Ye, Z.-X., Chen, Q., Wang, W., & Ling, Z.-H. (2020). Align, mask and select: A simple method for incorporating commonsense knowledge into language representation models.

Yih, W.-t., Chang, M.-W., He, X., & Gao, J. (2015). Semantic parsing via staged query graph generation: Question answering with knowledge base. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 1321–1331. https://doi.org/10.3115/v1/P15-1128

Yu, J., Moosavi, N. S., Paun, S., & Poesio, M. (2021). Stay together: A system for single and split-antecedent anaphora resolution. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4174–4184. https://doi.org/10.18653/v1/2021.naacl-main.329

Yu, L., & Ettinger, A. (2020). Assessing phrasal representation and composition in transformers. In B. Webber, T. Cohn, Y. He, & Y. Liu (Eds.), *Proceedings of the 2020 conference on empirical methods in natural language processing, EMNLP 2020, online, november 16-20, 2020* (pp. 4896–4907). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.397

Yu, M., & Dredze, M. (2014). Improving lexical embeddings with semantic knowledge. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 545–550. https://doi.org/10.3115/v1/P14-2089

Yu, M., Yin, W., Hasan, K. S., dos Santos, C., Xiang, B., & Zhou, B. (2017). Improved neural relation detection for knowledge base question answering. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 571–581. https://doi.org/10.18653/v1/P17-1053

Yu, W., Zhu, C., Fang, Y., Yu, D., Wang, S., Xu, Y., Zeng, M., & Jiang, M. (2021). Dict-bert: Enhancing language model pre-training with dictionary. *arXiv preprint arXiv:2110.06490*.

Zaremba, W., Sutskever, I., & Vinyals, O. (2015). Recurrent neural network regularization.

Zelle, J. M., & Mooney, R. J. (1996). Learning to parse database queries using inductive logic programming. *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, 1050–1055.

Zeng, C., Li, S., Li, Q., Hu, J., & Hu, J. (2020). A survey on machine reading comprehension: Tasks, evaluation metrics and benchmark datasets.

Zeng, D., Liu, K., Lai, S., Zhou, G., & Zhao, J. (2014). Relation classification via convolutional deep neural network. *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2335–2344. https://aclanthology.org/C14-1220

Zettlemoyer, L., & Collins, M. (2007). Online learning of relaxed CCG grammars for parsing to logical form. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 678–687. https://aclanthology.org/D07-1071

Zettlemoyer, L. S., & Collins, M. (2005). Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, 658–666.

Zha, H. (2002). Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 113–120. https://doi.org/10.1145/564376.564398

Zhai, Z., Nguyen, D. Q., Akhondi, S., Thorne, C., Druckenbrodt, C., Cohn, T., Gregory, M., & Verspoor, K. (2019). Improving chemical named entity recognition in patents with contextualized word embeddings. *Proceedings of the 18th BioNLP Workshop and Shared Task*, 328–338. https://doi.org/10.18653/v1/W19-5035

Zhang, L. (2002). Knowledge graph theory and structural parsing. *Chemical Engineering Science - CHEM ENG SCI*.

Zhang, M., Zhang, Y., & Fu, G. (2017). End-to-end neural relation extraction with global optimization. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1730–1740.

Zhang, M., & Qian, T. (2020). Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 3540–3549. https://doi.org/10.18653/v1/2020.emnlp-main.286

Zhang, S., Tong, H., Xu, J., & Maciejewski, R. (2019). Graph convolutional networks: A comprehensive review. *Computational Social Networks*, *6*, 1–23.

Zhang, Y., Yu, X., Cui, Z., Wu, S., Wen, Z., & Wang, L. (2020). Every document owns its structure: Inductive text classification via graph neural networks. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 334–339. https://doi.org/10.18653/v1/2020.acl-main.31

Zhang, Y., Zhong, V., Chen, D., Angeli, G., & Manning, C. D. (2017). Position-aware attention and supervised data improve slot filling. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 35–45. https://doi.org/10.18653/v1/D17-1004

Zhang, Y., Qi, P., & Manning, C. D. (2018). Graph convolution over pruned dependency trees improves relation extraction. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2205–2215. https://doi.org/10.18653/v1/D18-1244

Zhang, Y., Dai, H., Kozareva, Z., Smola, A. J., & Song, L. (2017). Variational reasoning for question answering with knowledge graph.

Zhao, K., Xu, H., Cheng, Y., Li, X., & Gao, K. (2021). Representation iterative fusion based on heterogeneous graph neural network for joint entity and relation extraction. *Knowledge-Based Systems*, 106888. https://doi.org/https://doi.org/10.1016/j.knosys.2021.106888

Zheng, Z., Si, X., Li, F., Chang, E. Y., & Zhu, X. (2012). Entity disambiguation with freebase. *The 2012 IEEE/WIC/ACM International Conference on Web Intelligence (WI'2012)*.

Zhong, V., Xiong, C., & Socher, R. (2017). Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*.

Zhou, B., Ning, Q., Khashabi, D., & Roth, D. (2020). Temporal common sense acquisition with minimal supervision. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7579–7589. https://doi.org/10.18653/v1/2020.acl-main.678

Zhou, H., Young, T., Huang, M., Zhao, H., Xu, J., & Zhu, X. (2018). Commonsense knowledge aware conversation generation with graph attention. *IJCAI*.

Zhou, J., Naseem, T., Fernandez Astudillo, R., Lee, Y.-S., Florian, R., & Roukos, S. (2021). Structure-aware fine-tuning of sequence-to-sequence transformers for transition-based AMR parsing. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 6279–6290. https://doi.org/10.18653/v1/2021.emnlp-main.507

Zhou, J., Zhang, Z., Zhao, H., & Zhang, S. (2020). Limit-bert : Linguistic informed multi-task bert.

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., & Xu, B. (2016a). Attention-based bidirectional long short-term memory networks for relation classification. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 207–212. https://doi.org/10.18653/v1/P16-2034

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., & Xu, B. (2016b). Attention-based bidirectional long short-term memory networks for relation classification. *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*, 207–212.

Zhu, J., Xia, Y., Wu, L., He, D., Qin, T., Zhou, W., Li, H., & Liu, T.-Y. (2020). Incorporating bert into neural machine translation.

Zipf, G. K. (1932). *Selected studies of the principle of relative frequency in language*. Harvard University Press. https://doi.org/doi:10.4159/harvard.9780674434929