Proceedings of the 6th International Workshop on Computational Terminology (COMPUTERM 2020), pages 26–36
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

Which Dependency Parser to Use for Distributional Semantics in a Specialized Domain?

Pauline Brunet[1], **Olivier Ferret**[1], **Ludovic Tanguy**[2]
1. CEA, LIST, F-91191 Gif-sur-Yvette, France
2. CLLE: CNRS & University of Toulouse, France
{pauline.brunet,olivier.ferret}@cea.fr
ludovic.tanguy@univ-tlse2.fr

Abstract

We present a study whose objective is to compare several dependency parsers for English applied to a specialized corpus for building distributional count-based models from syntactic dependencies. One of the particularities of this study is to focus on the concepts of the target domain, which mainly occur in documents as multi-terms and must be aligned with the outputs of the parsers. We compare a set of ten parsers in terms of syntactic triplets but also in terms of distributional neighbors extracted from the models built from these triplets, both with and without an external reference concerning the semantic relations between concepts. We show more particularly that some patterns of proximity between these parsers can be observed across our different evaluations, which could give insights for anticipating the performance of a parser for building distributional models from a given corpus.

Keywords: Dependency parsing, distributional semantics, specialized corpus, biomedical domain

1. Introduction

This work takes place in the broader context of studying distributional semantic analysis methods for specialized corpora. This type of corpora are usually small-sized (a few million words or less), which poses a challenge for distributional methods, and contain specific, highly technical vocabulary, meaning that adapting methods based on large generic corpora might be difficult. We make the hypothesis, supported by the work of (Tanguy et al., 2015), that the small amount of data may be circumvented by a method based on syntactic contexts. Such methods have already been investigated by a large body of work. The largest part of it is dedicated to count-based approaches (Grefenstette, 1994; Habert et al., 1996; Lin, 1998; Curran and Moens, 2002; Padó and Lapata, 2007; Baroni and Lenci, 2010) but it also includes work adding dimensionality reduction methods (Lapesa and Evert, 2017) or more recently, work about word embeddings (Levy and Goldberg, 2014). One of our focuses is to select the best-suited tools for semantic analysis of specialized corpora. In particular, given that syntactic contexts will be a building block for the task, which syntactic parser should be used to extract these contexts? The goal of this article is, thus, to study the impact of the choice of parser on the construction of a distributional model with a frequency-based method. Our work is not the first work on comparing different parsers. Several evaluation campaigns were previously organized for various languages: the Easy (Paroubek et al., 2008), Passage (De La Clergerie et al., 2008), SPMRL (Seddah et al., 2013) and CoNLL (Zeman et al., 2018) campaigns as well as more focused studies like (Candito et al., 2010) or (De La Clergerie, 2014). However, the benchmarks used in these studies, adopting the kind of diverse, generic corpora on which the tools have been trained, might not be the most relevant option for specialized corpus parsing. Moreover, even though some of these campaigns are recent, the main tools available have not been compared on the same evaluation sets. We previously performed a first study (Tanguy et al., 2020), comparing 11 different versions of parsers on a small specialized corpus made up of Natural Language Processing papers for French. However, we lacked a reliable external reference to measure the results of the parsers against. So our evaluation was only a qualitative comparison.

2. Overview

To go beyond the limitation in (Tanguy et al., 2020), we have chosen, in the work we present in this article, to run a new evaluation on a small, specialized biomedical corpus, whose building is described in Section 3.1 and for which we may compare the relations implied by the extracted syntactic contexts against an external resource, the Unified Medical Language System (UMLS) (Bodenreider, 2004), which contains relations between medical and biomedical concepts (see Section 3.3).

More precisely, we defined the following process: we applied each of the 10 studied parsers we present in Section 3.2 to the corpus, outputting morphological, syntactic and grammatical information. In parallel, we ran MetaMap (Aronson and Lang, 2010), a biomedical entity linker, to identify biomedical concepts as defined and recorded in the UMLS. Then, we aligned these concepts with the tokens outputted by the parsers (see Section 4.1). From this alignment, we were able to extract grammatical relations between concept-mapped tokens and other tokens, which gave us syntactic contexts for concept-mapped tokens, and, therefore, for biomedical concepts themselves (see Section 4.2). We then built distributional thesauri for each of the parsers (see Section 5.1), leading to a large set of distributional similarity relations between biomedical concepts. Finally, we compared these similarity relations to the relations between biomedical concepts given by the UMLS (see Section 5.3) and used this comparison for characterizing our studied parsers.

3. Experiment Framework

3.1. Corpus

For this experiment, we used a small part of the Open Access subset of the PubMed Central corpus (PMC)[1], a collection of more than 5 million full-text articles from thousands of biomedical and life science journals. This corpus, originally in a very rich XML format, was cleaned up by removing a lot of non-parsable content like tables, formulas, links, then converted to raw text for parsing. We chose a subset based on a specialty domain centered on stem cells. Articles in PMC OA are indexed by the MeSH index, which tags each article with their themes (or subject headings), with an indication of whether the theme is a main theme of the article or not. To obtain a corpus that was the right size for our purposes, we chose to include any article that was tagged with a heading containing the words ”Stem Cells“, which includes headings such as ”Stem Cells“, ”Adult Stem Cells“, ”Totipotent Stem Cells“, ”Mouse Embryonic Stem Cells“, and others. This was done regardless of whether the heading was indicated as a main theme of the article or not. The resulting corpus is comprised of 23,094 articles, and 104 million words.

3.2. Syntactic Parsers

We selected 5 tools able to perform dependency parsing in English, focusing on easily available and ready-to-use parsers, i.e. those that take in charge the whole processing chain, from raw text to dependencies. These tools were applied with their default options.

All these tools use statistical models trained on annotated corpora. Their differences concern implementation choices like parsing techniques (graph- or transition-based, for instance), learning models (SVM, maximal entropy or more recently, recurrent neural networks), and upstream or side processing (segmentation, lemmatization). There is much less choice among the training corpora, given the high cost of the annotation and validation processes.

CoreNLP (Manning et al., 2014), the main tool of the Stanford team, implements a maximum entropy tagger, which uses the Penn Treebank tagset (Marcus et al., 1993), and a transition-based parser.

StanfordNLP (Qi et al., 2018) is a tool that, on top of giving access to the CoreNLP chain in Python, implements an entirely different parsing chain. Its graph-based parser relies on a LSTM neural network. StanfordNLP offers 3 English models, trained on the UD **EWT** (Silveira et al., 2014), **LinES** (Ahrenberg, 2015) and **ParTUT** (Bosco et al., 2012) corpora. We used all three of these models.

Spacy is an industry-targeting tool whose main characteristic is its speed compared to most other parsers. The tagger is based on a perceptron, with attributes based on Brown clusters, following (Koo et al., 2008). It implements a non-monotonous transition-based parser which can revise previous decisions (Honnibal and Johnson, 2015). The default model we used was trained on OntoNotes (Hovy et al., 2006) and uses the ClearNLP dependency labels[2].

UDPipe (Straka and Straková, 2017) uses a neural network with a Gated Recurrent Unit mechanism to do both tokenization and segmentation at once. For PoS tagging, it generates possible tags for words from their suffix and performs disambiguation with a perceptron. The transition-based parsing relies on a simple one-layer neural network. UDPipe includes four English models. We used all of them, trained on the UD **GUM** (Zeldes, 2017), **EWT**, **LinES** and **ParTUT** corpora.

Talismane (Urieli and Tanguy, 2013) uses a mix of statistic models and language-specific features and rules incorporating linguistic knowledge. It was trained on the Penn Treebank.

We are fully aware that these parsers can only be compared on a practical level since the technologies used, their goals, their training data, and even the times at which they were created can scarcely be compared.

3.3. Terminological Reference Resource

The UMLS is a set of knowledge sources related to biomedical sciences. The main part of the system is the UMLS Metathesaurus, which aggregates nearly 200 biomedical controlled vocabularies in an attempt to provide a reference frame for medical and biomedical concepts and links the different names under which they are known in different vocabularies as synonyms. The Metathesaurus is organized around these concepts, which, in theory, have only one meaning, and are unique in the Metathesaurus. Each concept has a unique identifier called CUI and is linked to one or more names, in specific vocabularies, for this concept, which have identifiers called AUI.

For example, the concept ”Headache“ (CUI: C0018681) can be found as the following variations (among others): in vocabulary SNOMED, ”Headache“ (AUI: A2882187), in vocabulary MeSH, ”Headache“ (AUI: A0066000) and ”Cranial Pain“ (AUI: A1641293), and in vocabulary DxP, ”HEAD PAIN CEPHALGIA“ (AUI: A0418053).

On top of these concepts, the Metathesaurus provides some relations between concepts[3]. Most of these relations come from individual source vocabularies; some of these are added by the Metathesaurus maintainers and the others by the users.

All relations have general REL labels, which specify the type of relations: synonym, parent, child, sibling, broader, narrower, qualifier, qualified by, or unspecified (several degrees). There are 14 possible REL labels.

Around one-fourth of relations also have a RELA label, which further specifies the relation, like is_a, has_ingredient, property_of. . . These labels come from the source vocabularies. As such, they are more diversified

[1] http://www.ncbi.nlm.nih.gov/pmc

[2] https://github.com/clir/clearnlp-guidelines/blob/master/md/specifications/dependency_labels.md

[3] https://www.ncbi.nlm.nih.gov/books/NBK9684/#_ch02_sec2_4_

than the REL labels, with nearly 900 different labels in total, and we cannot assume that they are coherently used throughout the Metathesaurus.

We observed that the distribution of relations among concepts (CUI) is not very well balanced:

	number of relations / CUI
mean	23.7
standard deviation	160.0
median	6.0
max	14,112
min	1

Some concepts have a very large number of relations but most of them are only linked to a restricted number of other concepts. As a consequence, since our objective in this study is to characterize the distributional neighborhood of the largest number of concepts as possible, we chose to keep for our evaluation as many reference relations as possible and not to select them according to their type in the UMLS Metathesaurus.

The only selection was performed in relation to our target domain. We had no indication in the Metathesaurus for selecting the relations specifically tied to the domain of stem cells. Hence, this selection was done indirectly by keeping only the relations between the concepts identified in our corpus for this domain. In practice, starting from an initial set of 112,790 concepts linked by 2,845,112 relations for the whole Metathesaurus, we obtained a set of 45,762 concepts linked by 1,272,224 relations. The selection rate is quite comparable for concepts and relations – 40.6% for concepts and 44.7% for relations – and the distribution of the number of relations by concept after this selection is close to that for the whole Metathesaurus:

	number of relations / CUI
mean	24.6
standard deviation	135.9
median	7.0
max	8,338
min	1

4. From Corpus to Dependency Triples

4.1. Concept Identification

The first step in our study is to match tokens, as segmented by various parsers, to biomedical concepts, as recognized by a biomedical entity linking tool. This task was greatly impeded by various alignment issues.

There are some available tools for biomedical UMLS concepts extraction from text. After testing several of them, among which cTakes (Savova et al., 2010) and QuickUMLS (Soldaini and Goharian, 2016), we decided to use MetaMap (Aronson, 2001), the reference tool for this task, because it had the clear advantage of providing disambiguation between possible candidate concepts for a phrase.

MetaMap splits the input documents into sentences, which are further split into phrases. It analyzes these phrases individually and outputs candidate mappings of UMLS concepts to the phrase. These mappings are given an evaluation score based on 4 metrics: centrality, variation, coverage, and cohesiveness. The mapping with the highest score may be selected as the most likely to be correct but MetaMap also provides a disambiguation module based on context. We exploited the possibility of MetaMap to output only the most likely mapping based on score and context disambiguation.

For instance, the phrase ”Generation of single-copy transgenic mouse embryos“ is linked to UMLS concepts ”Generations“ (C0079411), "Singular“ (C0205171), ”Copy“ (C1948062), ”Mice, Transgenic“ (C0025936) and ”Embryo“ (C0013935).

The linguistic analysis performed by MetaMap for identifying concepts in documents is, of course, different from the analysis performed by our target parsers. More precisely, the tokenization step is particularly important for aligning the concepts it identifies with the tokens issued from the various tokenizations of our parsers. MetaMap gives two position data for each match. The first one gathers the character offset in the phrase and the length of the matched words. This information is highly difficult to match with parsers' offsets because of imprecisions and different counting conventions from both MetaMap and the various parsers.

The second position information is the rank of the matched words in the phrase. For instance, in the above example, the concept ”Generations“ has both a *TextMatchStart* and *TextMatchEnd* attributes equal to 1 while ”Mice, Transgenic“ has a *TextMatchStart* attribute of 5 and a *TextMatchEnd* attribute of 6. However, this information cannot be directly matched with the tokenization of a parser since it depends on both the tokenization and MetaMap's phrase splitting.

The first step is then to associate each concept identified by MetaMap with its own tokenization, which we later align with each parser's tokenization.

4.1.1. Matching MetaMap's tokens with MetaMap's concepts

This first step is not trivial as the tokenization performed by MetaMap is not directly accessible in its output. However, each phrase in this output is segmented into syntactic units and each of these units is associated with a list of tokens. For example, ”single-copy“ is a syntactic unit associated with the token list [”single“, ”copy“]. From these syntactic units, we can collect the associated tokens and number them according to their order, which gives us something close to the behind-the-scene MetaMap tokenization. This numbering can be used to match the tokens with the biomedical concepts, but with the necessity to take two additional problems into account. First, the punctuation is not considered in MetaMap's numbering, but must obviously be recorded for the later alignment with the parsers' tokenization. Second, MetaMap's numbering sometimes skips the first or the first few tokens in a phrase if they are not associated with a concept, which is more troublesome. For example, in the phrase ”from cells“, MetaMap may declare that the ”Cells“ concept starts from 1 or 2. We were not able to determine the cause of this behavior but we found a workaround for the problem by comparing the offset of the matched starting position to the offset of the phrase starting position. If a discrepancy was found, the character count of each word was added until the discrepancy was filled to compute the number of skipped words.

With this process, we were able to match MetaMap's concepts to its tokenization with very good accuracy.

4.1.2. Matching MetaMap's tokens with parsers' tokens

The next step matches MetaMap's tokenization with each parser's tokenization. Another solution could have consisted in feeding MetaMap's tokenization to the parsers, as most of them are modular enough to allow it. We rejected the idea for two reasons. First, the tokens we retrieved from MetaMap could be different from the initial text: for example, by modifying some punctuations, destroying case information and even expanding acronyms. Second, we wanted to use the parsers out of the box, with their own tokenization suited to their own tagging and parsing processes.

The algorithm for matching these different tokenizations is based on the fact that the tokenizations are essentially similar. The majority of words are tokenized similarly by MetaMap and the parsers. Thus, for each document, we can align the outputs of MetaMap and the considered parser by relying on their common tokens and use a small set of heuristics to deal with discrepancies. The discrepancies we handle are of several types. They may come from parser-specific issues, such as Spacy inserting ”SPACE“ tokens when confronted with large breaks in the text. One of the two tokenizations may have inserted a sentence break while the other may not, in which case the sentence break is skipped. One of the tokenizations may have split a token while the other may not, such as ”single-copy“ on one side and ”single“ followed by ”copy“ on the other side. In such cases, we add the next tokens on the shorter side, separating them with both spaces and ”-“ until they stop matching the longer side or the whole split token is covered. If the process has been successful, the longer token is matched with all the smaller ones. If it has failed, we skip tokens on both sides and see if the next ones match. This is especially useful for cases where MetaMap or the parser modifies the tokens in some way, like ”99%“ becoming ”99“. Failing that, we are only concerned with finding some part of the text where the tokens match again, ideally as close as possible to the failure point. We implement this strategy by recording, from the failure point, the list of tokens from both MetaMap and the target parser and checking at each step if the last two tokens seen on one side can be found in the list of the other side[4]. If so, we skip up to this part and start matching from there.

This algorithm works fairly well and a very large percentage of tokens are matched.

4.2. Dependency Triple Extraction

The next step is to extract dependency relations between words to build the contexts that will be used for distributional analysis. This follows a similar process to the work of (Lin, 1998) and produces, from the dependency relations outputted by syntactic parsers, typically illustrated by Figure 1, the representation of the contexts of a word in a corpus under the form of syntactic triplets (dependent, relation, governor).

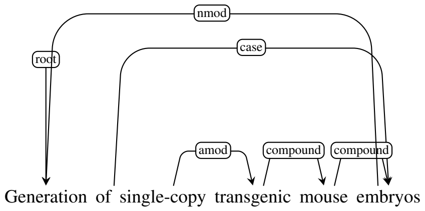

Figure 1: Dependencies identified by the Stanford NLP parser trained on the LinES corpus for the phrase ”Generation of single-copy transgenic mouse embryos“.

Not all relations provide useful context information. Generally, relations including closed-class words (determiners, conjunctions, pronouns, etc.) are not considered for building distributional contexts. For this study, we performed our selection not on the PoS tag of the governor and dependent, but on the dependency relation itself, choosing to exclude some of them.

For parsers following the Universal Dependency scheme, the excluded relations were `root`, `cc`, `cc:preconj`, `punct`, `case`, `mark`, `det`, `det:predet`, `cop`, `neg`, `aux`, and `nmod:poss`. Typically, relations such as `neg`, `aux` or `det` include negation markers (*not*...), auxiliary verbs (*have*, *be*, *can*:.) or determiners (*the*, *a*...) that we don't want to see in distributional contexts. For Spacy, it was `root`, `ccc`, `case`, `prep`, `det`, `neg`, `expl`, `predet`, `aux`, `auxpass`, and `mark`. For Talismane, given its specific dependency scheme, we had to rely on PoS tags for achieving the same kind of filtering, excluding IN, DT, MD, CC, EX, PDT, PRP, PRP$, TO and, RP.

Relations with prepositions had to be modified to link the actual related words and include the preposition in the relation. We illustrate the different ways this kind of grammatical constructions are parsed with the phrase ”region within the cluster“. Figure 2 shows the output produced by UD dependency scheme-following parsers.

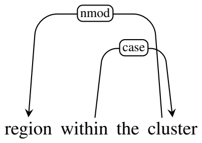

Figure 2: Dependencies identified by Universal Dependencies scheme-type parsers for the phrase ”region within the cluster“.

Figure 3 gives the result of the parsing by Spacy.

Finally, Figure 3 presents the output of Talismane for this phrase.

These three cases are the three basic patterns of how prepositions are managed in each scheme. We normalize all these

[4] We tested the use of one token instead of two but found better results with two tokens.

[6] `http://www.mathcs.emory.edu/~choi/doc/cu-2012-choi.pdf`

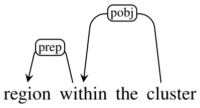

Figure 3: Dependencies identified by SpaCy (Clear Style dependencies)[6]for the phrase ”region within the cluster“.

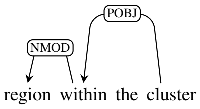

Figure 4: Dependencies identified by Talismane (Penn Treebank dependencies) for the phrase ”region within the cluster“.

variants with a `prep/within` dependency relation, as illustrated in Figure 5.

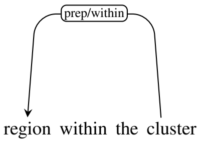

Figure 5: Dependency constructed from the existing dependencies for the phrase "region within the cluster“.

For the example of Figure 1, the resulting list of triplets is:

(embryos, prep/of, generation)
(generation, prep/of-1, embryos)
(single-copy, amod, transgenic)
(transgenic, amod-1, single-copy)
(transgenic, compound, mouse)
(mouse, compound-1, transgenic)
(mouse, compound, embryo)
(embryos, compound-1, mouse)

However, we need to adapt this representation of context to our task, which is specifically to extract the contexts of biomedical concepts. Thus, we only extract the relations where at least one side, the dependent or the governor, is a token that is part of a concept and the other side is not part of the same concept. Moreover, we only consider nominal concepts, which we define here as concepts where at least one word was tagged as a noun by the MetaMap tagger. Furthermore, for each side of a triplet, we include the following data:

- CUI: the unique UMLS id if it is a concept, _ otherwise;
- PREF: the preferred form of the concept in the UMLS if it is a concept, _ otherwise;
- NORM: the normalized form of the concept as it occurs in the text if it is a concept, _ otherwise. Concretely, it corresponds to the concatenation of all the lemmas of the concept;
- LEMMA of the token actually part of the relation;
- PoS of the token actually part of the relation.

In the above phrase, five concepts were recognized by MetaMap: ”Generation“ (noun), ”Singular“ (adj), ”Copy (object)“ (adj), ”Mice, Transgenic“ (noun), and ”Embryo“ (noun). The corresponding triplets are given in Table 1.

4.3. Comparison of Parsers in Terms of Dependency Triples

Several previous studies (starting with (Grefenstette, 1994) and (Lin, 1998)) have considered subsets of syntactic relations for distributional models. More recent works (Padó and Lapata, 2007; Baroni and Lenci, 2010) have selected a short list of core relations, and we decided to limit our experiment to these, which we regrouped in the categories described below. They follow the main syntactic relations identified by dependency parsers and correspond to the minimal configuration of (Padó and Lapata, 2007), to which we added the last one from (Baroni and Lenci, 2010), that consider the prepositions themselves as relations between a head and a dependent word, as described in section 4.2.

`N suj V`: nominal subject of a verb;

`N obj V`: nominal direct object of a verb;

`ADJ mod N`: adjective modifying a noun;

`ADV mod ADJ/V`: adverb modifying an adjective or a verb;

`X coord X`: coordination between two nouns, adjectives, adverbs, or verbs (note: the conjunction itself is not considered);

`X prep_P X`: prepositional binding between nouns, adjectives, or verbs.

This brings down the number of triplets (occurrences) extracted from each parser from around 60M to around 40M, with SpaCy having the least (38.3M) and the version of UDPipe trained on ParTUT having the most (52.4M, far ahead of the others).
We compare the triplets' coverage between parsers but first, we reduce the triplets to some core elements: the CUI of the left-hand side, the CUI or lemma (depending on whether it is a concept or not) and the PoS tag of the right-hand side, and the relation between the two.
Our triplets now look like the following:

C0040648	N:C0237753_prep/of
C1883221	ADJ:distinct_mod-1

We also limit ourselves to triplets in which the left-hand side (CUI) appears at least 10 times in each parser's output, and in which the right-hand side (CUI/lemma and PoS) appears at least twice in each parser's output. This results in 39M unique triplets, of which 21M appear for at least two parsers and 3.5M appear for all parsers. Among these 39M triplets, each parser has found around 13M of them, with the least being CoreNLP with 12.2M and SpaCy with 12.5M. The parser with the highest number of common triplets is the ParTUT version of UDPipe, with 16.8M triplets.

Dependent					Relation	Governor				
CUI_1	$PREF_1$	$NORM_1$	$LEMMA_1$	PoS_1		CUI_2	$PREF_2$	$NORM_2$	$LEMMA_2$	PoS_2
C0013935	Embryos	embryo	embryo	NOUN	prep/of	C0079411	Generations	generation	generation	NOUN
C0079411	Generations	generation	generation	NOUN	prep/of-1	C0013935	Embryos	embryo	embryo	NOUN
C0025936	Mice,Transgenic	transgenic_mouse	transgenic	NOUN	amod-1	_	_	_	single-copy	ADJ
C0025936	Mice,Transgenic	transgenic_mouse	mouse	NOUN	compound	C0013935	Embryos	embryo	embryo	NOUN
C0013935	Embryos	embryo	embryo	NOUN	compound-1	C0025936	Mice,Transgenic	transgenic_mouse	mouse	NOUN

Table 1: Syntactic triplets with concepts identified by MetaMap

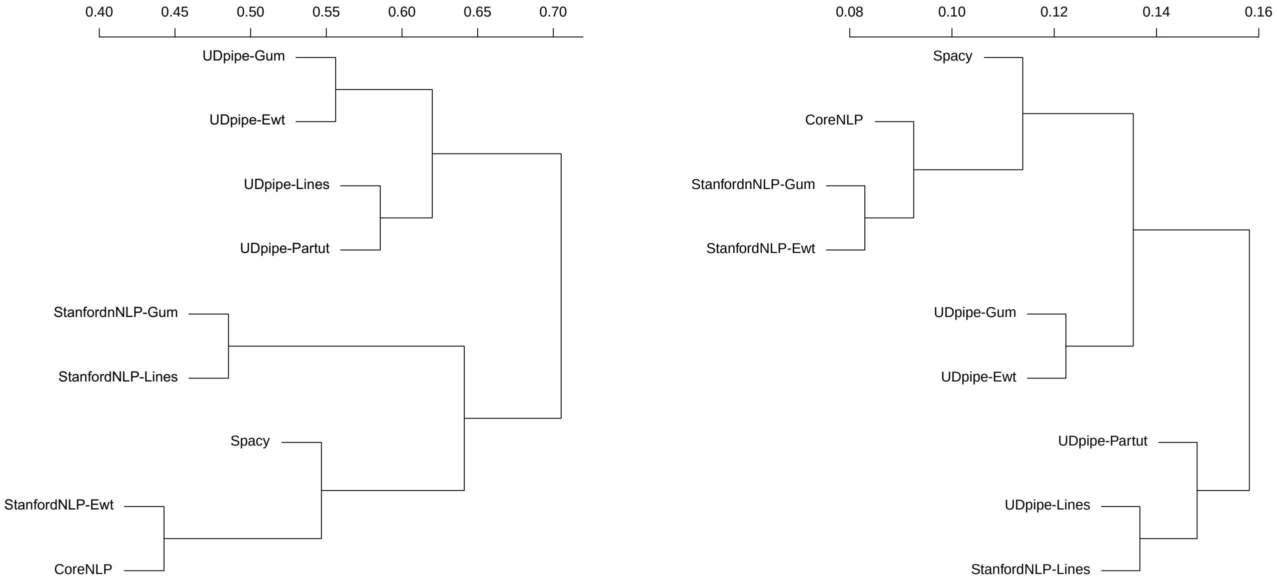

Figure 6: Hierarchical clustering of parsers according to their correlation on the triplets found by at least two parsers (right side) and all parsers (left side).

We computed the agreement of the parsers about the triplets they produced by computing Spearman's correlation coefficient (ρ) both for the 21M triplets shared by at least two parsers and the 3.5M triplets common to all parsers. The first measure focuses on the differences between parsers in terms of diversity of triplets while the second measure looks more precisely at their differences in terms of frequency for the common triplets. We did not include Talismane as it was difficult to adapt its PoS tags and dependency labels to our normalization, as was done for CoreNLP and SpaCy. Figure 6 shows the hierarchical clustering of the parsers according to these correlations (more precisely, 1 - ρ for having a distance).

Globally, we can observe that the type of parser has a significant impact on the triplets, which is not a surprise: the UDPipe parsers are particularly close to each other but most of the StanfordNLP models are also grouped. However, the training corpus can also have an impact when we consider the triplets shared by all parsers, with StanfordNLP-Lines much closer to UDpipe-Lines than to the two other StanfordNLP models. This is why the clusterings built for the two sets of triplets are a little bit different, even if they also share some patterns: for instance, SpaCy is close to CoreNLP, which is close to StanfordNLP-Ewt while UDpipe-Gum and UDpipe-Ewt form a group for the two sets.

5. Distributional Models

5.1. Building of Distributional Models

Following the distinction made in Baroni et al. (2014), we built our distributional models according to a count-based approach, such as in (Lin, 1998), rather than according to a predictive approach such as in (Mikolov et al., 2013). The first justification of this choice is that, except for (Levy and Goldberg, 2014), the number of studies relying on dependency relations is very limited among predictive approaches. More importantly, some recent studies (Pierrejean and Tanguy, 2018) have shown that predictive approaches are unstable to some extent concerning the search of the nearest distributional neighbors of a word. Since we want specifically to concentrate on the effects resulting from the use of different syntactic parsers, we adopted a count-based approach.

We implemented this approach by building on the findings of recent studies in the field (Kiela and Clark, 2014; Baroni et al., 2014; Levy et al., 2015) and more particularly took up two main options from (Ferret, 2010): the use of Positive Pointwise Mutual Information (PPMI) for weighting the context elements and the application of very loose filtering that removes the elements of these contexts with only one occurrence. The second choice is justified by both the fairly small size of our target corpus and the experiments of (Ferret, 2010) with linear co-occurrents. The main particularity of our work is the fact that the entries of our distri-

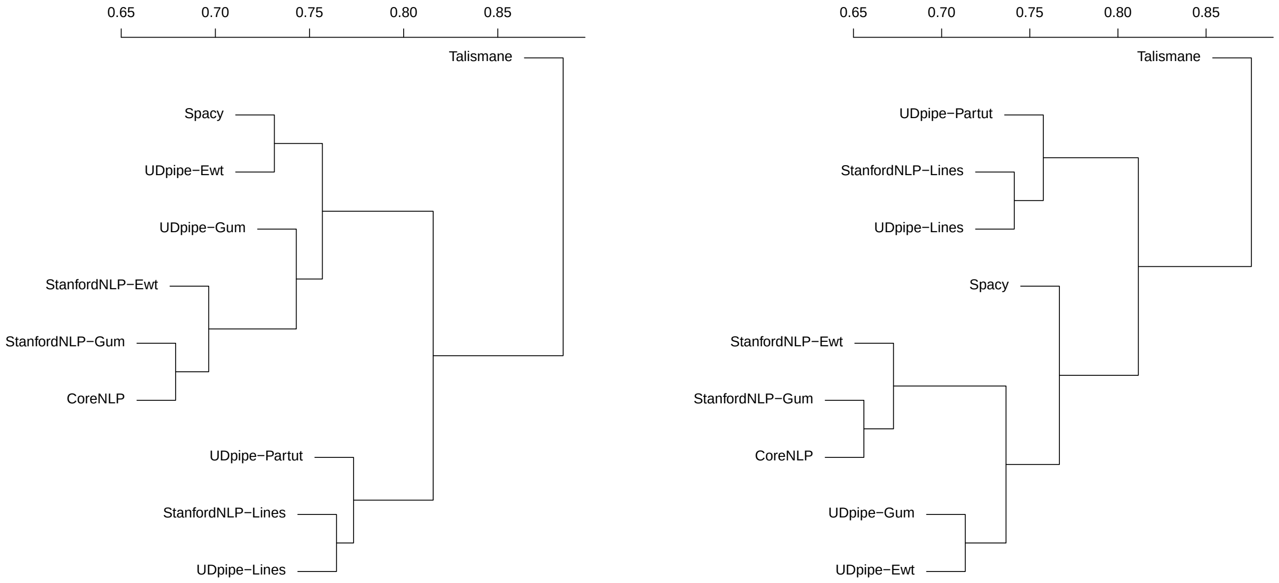

Figure 7: Hierarchical clustering of models according to their agreement on the first nearest neighbor (left side) and according to the RBO measure (right side).

butional models are not words but UMLS concepts. More precisely, each entry is made of the triple (CUI, PREF, PoS) under the form PREF_CUI#PoS. The elements of contexts can be either words or concepts since dependency triples can include concepts or words.

However, this particularity did not influence on the way we built our distributional models and we classically computed the similarity of two concepts by measuring the Cosine similarity score between their contexts vectors. For a given model, this computation was done for each pair of concepts with contexts sharing at least one element. The results of this process can also be viewed as a distributional thesaurus in which each entry corresponds to a concept of the considered vocabulary and is associated with the list of all other concepts of this vocabulary, sorted in descending order of their similarity value with the entry. In practice, only the nearest 100 distributional neighbors are kept, which is a fairly large number compared to the average number of relations by concept – 24.6 – but is justified by the fact that some concepts may have a much higher number of relations.

5.2. Comparison of Distributional Models

The first step for comparing our distributional models, and more indirectly the parsers used for extracting the distributional data they rely on, was to compute their agreement of our models on the nearest neighbors retrieved for each word. Among the concepts shared by all models, 47,647 concepts had at least one distributional neighbor. For each pair of models, the agreement on the nearest neighbor retrieved for each concept was computed[7] and used for building a similarity matrix. Hierarchical clustering was performed from this matrix, which leads to the left side of Figure 7. First, we can observe that the model built from Talismane is clearly aside from the others. The second main trend is that the training corpus of the parsers can be more important than the type of parser. For instance, the StanfordNLP and UDPipe parsers trained on the LinES corpus are grouped together and fairly distant from the same parsers trained on the GUM and EWT corpora. However, among the parsers trained on these two corpora, which are fairly heterogeneous compared to the LinES corpus, the proximity between the models they contributed to build is guided by the type of parser.

Even if the overall aspect of the dendrogram is a little bit different due to the position of the model built from Spacy, these trends are globally confirmed by comparing the neighbors of concepts by the means of the *Rank-Biased Overlap* measure (Webber et al., 2010), as illustrated by the right side of Figure 7. This measure is applied to all neighbors of our thesaurus' entries (100 neighbors in practice) and extends the notion of average overlap – the average of the overlap between two lists at different ranks – by decreasing the importance of overlap as the rank of the considered neighbors increases. As a consequence, nearest neighbors are given greater importance. This importance is defined by the p parameter, which can be interpreted as the probability, starting from the beginning of the list of neighbors, to continue to consider the following neighbors in the list. The value $p = 0.98$ used in our case means that the first 50 nearest neighbors of an entry account for around 85% of the evaluation. Figure 7 is based on the distance $1 - RBO$, which can be considered as a metric.

The clusterings of Figure 7 can also be compared to those of Figure 6: Talismane is absent from Figure 6 but has a very limited impact in Figure 7 since it is clearly distant from the other parsers. This comparison shows that the clustering based on the distributional neighbors is much closer to the clustering based on the triplets shared by all parsers than to the clustering based on the triplets shared by only two parsers. This suggests that the triplets of the first set are

[7] Ratio of the number of words sharing the same nearest neighbor to the size of the considered vocabulary.

Model	#concepts	#eval. concepts	#rel./ concept	Recall	R_{prec}	MAP	P@1	P@5	P@10	P@100
StanfordNLP-Ewt	49,002	42,340	25.4	4.9	3.5	3.0	9.4	5.4	3.9	1.3
CoreNLP	49,022	42,360	25.3	4.7	3.4	2.9	9.2	5.2	3.8	1.2
StanfordNLP-Gum	48,524	41,998	25.4	4.5	3.1	2.6	8.7	4.9	3.6	1.1
StanfordNLP-Lines	47,671	41,275	25.7	4.5	3.1	2.6	8.6	4.8	3.6	1.1
UDpipe-Ewt	47,883	41,366	25.6	4.5	3.1	2.6	8.5	4.8	3.5	1.1
Spacy	49,895	43,112	25.2	4.1	3.1	2.5	8.4	4.6	3.4	1.0
UDpipe-Gum	47,133	40,832	25.7	4.3	3.0	2.5	8.4	4.6	3.4	1.1
UDpipe-Partut	47,233	40,859	25.8	4.3	3.0	2.5	8.3	4.7	3.4	1.1
UDpipe-Lines	46,645	40,408	25.8	4.0	2.7	2.3	7.6	4.2	3.1	1.0
Talismane	48,411	41,812	25.3	3.2	2.2	1.9	6.1	3.3	2.4	0.8

Table 2: Evaluation of our distributional models with UMLS relations as reference (measures x 100).

globally more frequent than the triplets of the second set and can be used for having a first indication of the proximity of the distributional models built from them.

5.3. Evaluation of Distributional Models

The comparison of our distributional models according to the neighbors of their entries gives some insights about their proximity but no information about their relevance for representing the semantic relations in the target domain. This second type of evaluation has to rely on a reference resource accounting for these relations, which can be done in our case by exploiting the UMLS relations we have presented in Section 3.3.

More precisely, we adopted the evaluation framework proposed in (Ferret, 2010), based on the Information Retrieval paradigm: each entry of our models is considered as a query and the sorted list of its distributional neighbors as the list of retrieved documents. In this context, a neighbor is considered as relevant if the pair (entry, neighbor) corresponds to a UMLS relation[8]. As mentioned in Section 3.3, no restrictions are applied to the type of these reference relations for two main reasons. First, we wanted to have a large enough set of relations for making our evaluation as reliable as possible. Second, even at the first level, with the REL labels, the relation types are fairly fuzzy in their definition, which makes the selection of a specific type of relations difficult in practice.

For measuring the relevance of the neighbors of an entry according to the UMLS relations, we adopted the classical measures used in the Information Retrieval field: R-precision, MAP (Mean Average Precision) and precision at various ranks (P@r). R-precision (R_{prec}) is the precision after the first R neighbors were retrieved, R being the number of reference relations while Mean Average Precision (MAP) is the average of the precision values calculated each time a reference relation is found. All these measures are given for each of our distributional models with a scaling factor equal to 100 by the six last columns of Table 2. The second column of this table corresponds to the number of concepts in each model while the third column is the number of these concepts with at least one UMLS relation. The fourth column gives the average number of UMLS relations for a concept in a model and the fifth column provides the average percentage of these relations that are present in the first 100 neighbors of each concept.

The results of this evaluation lead to several observations. First, their overall level seems to be fairly low. However, this is not abnormal given the size of our corpus. For instance, Ferret (2010) reports a value of 7.7 for R_{prec} with his most complete reference (38.7 reference relations by entry on average) but with a corpus nearly four times the size of ours. We can also observe from the second column that using different syntactic parsers has a limited but not negligible influence on the number of concepts extracted from the corpus: the model based on UDpipe-Lines has 5% fewer concepts than the model based on StanfordNLP-Ewt. In terms of global trends, the first two models, StanfordNLP-Ewt and CoreNLP, are slightly better than a group of seven models with fairly close results while the last model is more clearly distant in terms of performance. This last observation is fully consistent with the separate position of the corresponding model in the dendrograms of Figure 7. More globally, similarities between models in Table 2 are consistent with their similarities in Figure 7, which suggests that even without an external reference, the distributional models can be compared in terms of semantic relevance by focusing on the neighbors retrieved for their entries. For instance, StanfordNLP-Ewt, CoreNLP, and StanfordNLP-Gum are close to each other in the two evaluations. This is also the case for UDpipe-Ewt, Spacy, and UDpipe-Gum. The main difference between the two evaluations concerns the relative importance of the training corpus and the type of the parser: in Table 2, the type of the parser seems to be the main factor while in Figure 7, the two factors are more intertwined.

Figure 8 gives a more global view of similarities between models according to the UMLS relations they retrieve by reporting the same type of analysis as Figure 7 but restricted to neighbors having a UMLS relation with their entry. This view confirms the main observations resulting from the analysis of Table 2. The model built with Talismane is significantly different from the others and the main patterns in terms of clustering are present, with a group made up of CoreNLP, StanfordNLP-Gum, and StanfordNLP-Ewt and a group with UDpipe-Gum and UDpipe-Ewt. As a con-

[8] More precisely, it means that the neighbor is part of a UMLS relation including both the entry and the neighbor.

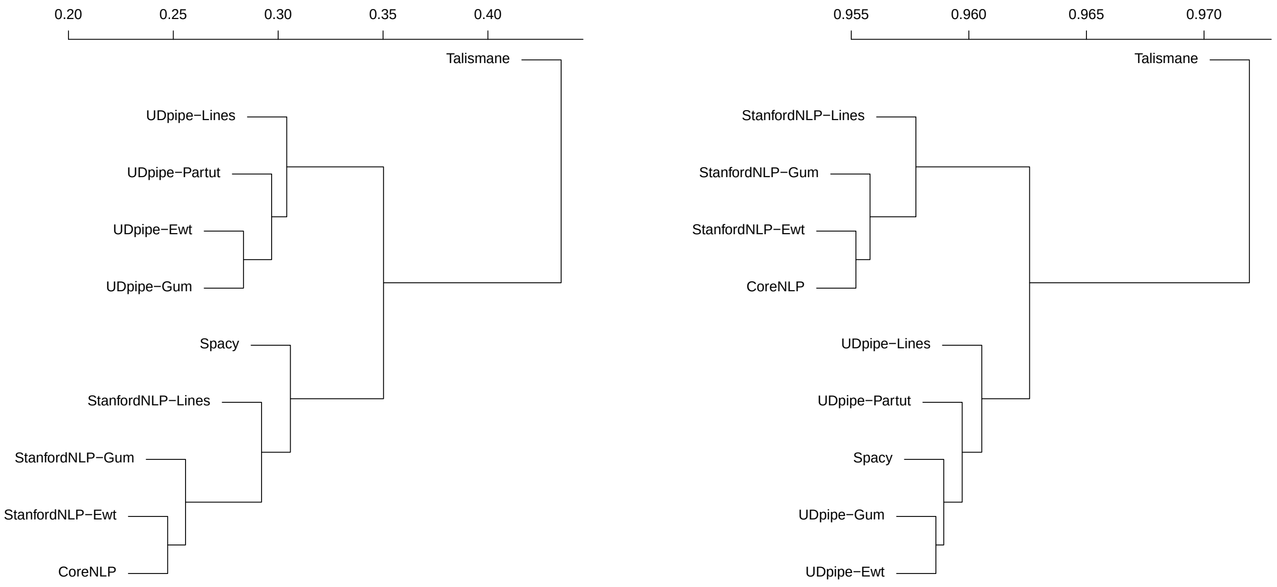

Figure 8: Hierarchical clustering of models according to their agreement on the first nearest neighbor with an UMLS relation with its entry (left side) and according to the RBO measure for all the neighbors having a UMLS relation with their entry (right side).

sequence, this evaluation emphasizes that the type of the parser used for extracting dependency triplets is the first criterion in terms of impact on the distributional models built from them but it also shows that in this context, the corpora used for training these parsers also have an influence and that heterogeneous corpora such as GUM and EWT are probably better for this training than a much more homogeneous corpus such as LinES.

6. Conclusion and Perspectives

In this article, we have investigated the influence of syntactic parsers on the distributional count-based models built from syntactic dependencies. More precisely, we have performed this study in the context of a specialized domain in the biomedical area with a moderate-size corpus made of scientific articles. One particularity of this study is to focus on the concepts of a reference ontology in the medical and biomedical areas. These concepts are mainly present in documents through multi-terms and identified by a reference tool, MetaMap, which requires aligning MetaMap's results with the results of the considered parsers. We have investigated the differences between parsers in terms of syntactic triplets but also in terms of distributional neighbors extracted from the models built from these triplets, both with and without an external reference concerning the semantic relations between concepts. We have more particularly shown the influence of the type of parser in these different evaluations but also the impact of the corpus used for training the parsers. Finally, we have found that some patterns of proximity between parsers are stable across our evaluations, which means that some measures applied to the output of syntactic parsers may perhaps be used to anticipate the performance of a parser for building distributional models from a given corpus. This will be the focus of our future work.

7. Acknowledgments

This work has been funded by French National Research Agency (ANR) under project ADDICTE (ANR-17-CE23-0001).

8. Bibliographical References

Ahrenberg, L. (2015). Converting an english-swedish parallel treebank to universal dependencies. In *Third International Conference on Dependency Linguistics (DepLing 2015), Uppsala, Sweden, August 24-26, 2015*, pages 10–19. Association for Computational Linguistics.

Aronson, A. R. and Lang, F.-M. (2010). An overview of metamap: historical perspective and recent advances. *Journal of the American Medical Informatics Association (JAMIA)*, 17(3):229–236.

Aronson, A. R. (2001). Effective mapping of biomedical text to the umls metathesaurus: the metamap program. In *Proceedings of the AMIA Symposium*, page 17. American Medical Informatics Association.

Baroni, M. and Lenci, A. (2010). Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721.

Baroni, M., Dinu, G., and Kruszewski, G. (2014). Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pages 238–247, Baltimore, Maryland.

Bodenreider, O. (2004). The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic acids research*, 32(suppl 1):D267–D270.

Bosco, C., Sanguinetti, M., and Lesmo, L. (2012). The parallel-TUT: a multilingual and multiformat treebank. In *Proceedings of LREC*, pages 1932–1938. European Language Resources Association (ELRA).

Candito, M., Nivre, J., Denis, P., and Anguiano, E. H. (2010). Benchmarking of statistical dependency parsers for French. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 108–116. Association for Computational Linguistics.

Curran, J. R. and Moens, M. (2002). Improvements in automatic thesaurus extraction. In *Workshop of the ACL Special Interest Group on the Lexicon (SIGLEX)*, pages 59–66, Philadelphia, USA.

De La Clergerie, E. V., Hamon, O., Mostefa, D., Ayache, C., Paroubek, P., and Vilnat, A. (2008). Passage: from French parser evaluation to large sized treebank. In *Proceedings of LREC*.

De La Clergerie, É. V. (2014). Jouer avec des analyseurs syntaxiques. In *Actes de TALN*.

Ferret, O. (2010). Testing semantic similarity measures for extracting synonyms from a corpus. In *7^{th} International Conference on Language Resources and Evaluation (LREC'10)*, pages 3338–3343, Valletta, Malta.

Grefenstette, G. (1994). *Explorations in automatic thesaurus discovery*. Kluwer.

Habert, B., Naulleau, E., and Nazarenko, A. (1996). Symbolic word clustering for medium-size corpora. In *16th International Conference on Computational Linguistics (COLING 1996)*, pages 490–495, Copenhagen, Denmark.

Honnibal, M. and Johnson, M. (2015). An Improved Non-monotonic Transition System for Dependency Parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378, Lisbon, Portugal, September. Association for Computational Linguistics.

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. (2006). Ontonotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL (HLT-NAACL 2006), Short Papers*, page 57–60, USA. Association for Computational Linguistics.

Kiela, D. and Clark, S. (2014). A Systematic Study of Semantic Vector Space Model Parameters. In *2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 21–30, Gothenburg, Sweden.

Koo, T., Carreras, X., and Collins, M. (2008). Simple semi-supervised dependency parsing. *Proceedings of ACL-08: HLT*, pages 595–603.

Lapesa, G. and Evert, S. (2017). Large-scale evaluation of dependency-based DSMs: Are they worth the effort? In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017), Short Papers*, volume 2, pages 394–400.

Levy, O. and Goldberg, Y. (2014). Dependency-Based Word Embeddings. In *52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pages 302–308, Baltimore, Maryland, June.

Levy, O., Goldberg, Y., and Dagan, I. (2015). Improving Distributional Similarity with Lessons Learned from Word Embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics*, pages 768–774. Association for Computational Linguistics.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Marcus, M., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *ICLR 2013, workshop track*.

Padó, S. and Lapata, M. (2007). Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.

Paroubek, P., Robba, I., Vilnat, A., and Ayache, C. (2008). EASY, Evaluation of Parsers of French: what are the results? In *Proceedings of LREC*.

Pierrejean, B. and Tanguy, L. (2018). Towards qualitative word embeddings evaluation: Measuring neighbors variation. In *Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 32–39.

Qi, P., Dozat, T., Zhang, Y., and Manning, C. D. (2018). Universal Dependency Parsing from Scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170, Brussels, Belgium, October. Association for Computational Linguistics.

Savova, G. K., Masanz, J. J., Ogren, P. V., Zheng, J., Sohn, S., Kipper-Schuler, K. C., and Chute, C. G. (2010). Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513.

Seddah, D., Tsarfaty, R., Kübler, S., Candito, M., Choi, J. D., Farkas, R., Foster, J., Goenaga, I., Gojenola Galletebeitia, K., Goldberg, Y., Green, S., Habash, N., Kuhlmann, M., Maier, W., Nivre, J., Przepiórkowski, A., Roth, R., Seeker, W., Versley, Y., Vincze, V., Woliński, M., Wróblewska, A., and Villemonte de la Clergerie, E. (2013). Overview of the SPMRL 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 146–182, Seattle, Washington, USA, October. Association for Computational Linguistics.

Silveira, N., Dozat, T., de Marneffe, M.-C., Bowman, S., Connor, M., Bauer, J., and Manning, C. D. (2014). A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Soldaini, L. and Goharian, N. (2016). Quickumls: a fast, unsupervised approach for medical concept extraction. In *MedIR workshop, sigir*, pages 1–4.

Straka, M. and Straková, J. (2017). Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada, August. Association for Computational Linguistics.

Tanguy, L., Sajous, F., and Hathout, N. (2015). Évaluation sur mesure de modèles distributionnels sur un corpus spécialisé: comparaison des approches par contextes syntaxiques et par fenêtres graphiques. *Traitement automatique des langues*, 56(2).

Tanguy, L., Brunet, P., and Ferret, O. (2020). Extrinsic evaluation of french dependency parsers on a specialised corpus: comparison of distributional thesauri. In *12th International Conference on Language Resources and Evaluation (LREC 2020)*, Marseille, France.

Urieli, A. and Tanguy, L. (2013). L'apport du faisceau dans l'analyse syntaxique en dépendances par transitions : études de cas avec l'analyseur Talismane. In *Actes de TALN*, pages 188–201, Les Sables d'Olonne, France.

Webber, W., Moffat, A., and Zobel, J. (2010). A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)*, 28(4):1–38, November.

Zeldes, A. (2017). The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Zeman, D., Haji, J., Popel, M., Potthast, M., Straka, M., Ginter, F., Nivre, J., and Petrov, S. (2018). CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies. *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21.