Context Similarity and Semantic Relationships

UE 805 Supervised Project

Final Report

Written by

Nami Akazawa and **Emre Canbazer**

Supervised by

Sylvain Pogodalla

Reviewed by

Marie-laurence Knittel, Etienne Petitjean, and **Samantha Ruvoletto**

MSc Natural Language Processing

Academic Year 2020-2021

Abstract

In the field of natural language processing and cognitive science, vector representations of words (vector space models) play an increasingly important role. While much research is being conducted on vector space models, there are still questions concerning what information in text impacts the encoding of semantic relatedness in vector space models.

Throughout this report, we will first provide some context about word vector space models and the methodological techniques used for building them. We will also review previous research on building these types of models, especially those that incorporate lexical relation. To describe relation between the in a language, we make use of linguistic theories that introduce defined notions that give description and systematization of semantic relationships. We attempt to describe the semantic relation between the taret words and the similar word candidates through lexical functions. In the latter part of this report, we explain our approach to constructing vector space models using MANGOES software. We have tested with various experimental settings by manually inspecting each built embedding with a set of collocation. The results showed that the use of specific linguistic features extracted from the corpus produce slightly different embeddings and it is possible to retrieve collocation types by combining context parameter settings. However it is not possible to target only the extraction of syntagmatic relations.

Contents

1 Introduction

There are various different methods for representing words as vectors and various state-of-art distributional methods are applied to natural language processing (NLP) and cognitive science tasks such as automatic thesaurus extraction, information retrieval, and semantic priming to name few. These tasks benefit from a semantic vector space that can embed words and their features (such as the meaning of the word) and reflect relationship between words (e.g., similarities).

Research has shown that vector space models utilizing word co-occurrence are able to capture the representation of lexical meaning in words. Thus typically, semantic vector spaces are traditionally constructed from text by counting co-occurrence of a word with its context. An assumption of words that occur in the same contexts tend to have similar meanings is made by Harris (1954). One can define contexts as certain number of neighboring words of a target word or relation between words linked in a syntactic dependency.

Our project aims to explore the lexical relationships, if any, that can be captured by incorporating lexical and syntactic information to the distributional models. In particular, we are interested in using linguistic theories that explain and categorize semantic relatedness in languages.

This report aims to introduce the methodology of constructing and evaluating semantic space vector using dependency-based context and presenting our experiment results. In Section 2, we first summarize the concept of distributional semantics and then explain the distributional semantic method and syntax-based models. We want to relate context similarity obtained from distributional models to formal theories of semantic relatedness, thus we introduce various existing semantic relations. We describe in detail one linguistic theory, Meaning-Text Theory and especially it's explanatory combinatorial lexicology branch developed by Mel'čuk (2016), which provides a in-depth characterization of lexicographical definition and lexical functions. Several research aims to tackle collocation acquisition using embedding and exploiting the theory of lexical functions. Collocations such as *heavy rain* or *take [a] break*, are words or phrases that are often used with another word or phrase where one (the base) is freely chosen (i.e., *rain*, *break*) , while the choice of the other (collocate) is restricted (i.e., *heavy*, *take*), depending on the base.

Section 3 explains our experiment methods with various setting of dependency-based context and discusses our results. In the conclusion, we conclude with our findings and discuss possible future works.

This work is interested in finding how lexical relationships are expressed in automatically constructed distributional models from textual data and make some observation that can contribute to understand more about the nature of the semantic relatedness exist if any in an embedded space.

2 Background

2.1 Distributional Semantics

Developing from the distributional hypothesis proposed by Zellig Harris (Turney and Pantel, 2010), distributional semantics prioritizes the distributional method and uses vector spaces as a representation of its models. Distributional method—with the term 'distribution' indicating the set of contexts in which the target word is observed to occur (Clark, 2015)—suggests that its surroundings or context characterize the meaning of a word which can range from a few of words (to the right and to the left) to a paragraph or a whole document depending on the approach. An example of context is Firth's concept of collocations which has been important for computational linguistics since it emerged. What makes the Firthian notion of collocation unique is its strict independence from compositionality and the fact that it prioritizes the environment of the target word (and not its interior structure) to explain its behavior (Pulman, 2013), which serves to disambiguate the word meaning by taking into consideration its context.

2.1.1 Vector Space Model

Given the idea of the distributional semantics and its hypothesis, we will be assuming that every word can be represented as high-dimensional vectors in a common vector space. Research has shown the vector space can encode the meanings of words, and we can see the semantic relation of words using similarity or distance measures. The most simplistic construction of a vector space model (VSM) for words is that given a set of target words and a corpus, we define a set of basis elements. The basis elements can be a collection of unique words, lemmas, words with their part-of-speech tag or dependency relation. The number of the dimensions for the semantic space will thus be the number of the basis elements. Then, a target word's coordinates represent the frequency of each basis element occurring within a certain distance before or after the target word in the corpus.

One of the variations of the VSM we will be experimenting is the one in which the syntactic information of the target word and its surrounding text is taken into account and only the ones with certain relations are included (Clark, 2015; Padró et al., 2014; Heylen et al., 2008; Peirsman, Heylen, and Speelman, 2007).

2.1.2 Weighting Schemes

Depending on the number of the target words and basis elements, the co-occurrence matrix can be large or small. However, counting the raw frequency of words' co-occurrence is very skewed and non-discriminative. If we consider all the unique words in text as basis elements, rare word pairs will be overly infrequent and it will result in sparseness (0s in many cells) of the word-word matrix. One simple way to tackle this challenge is to have a frequency threshold to remove low frequency words. There also exists words such as $\langle$*the, of, as, and*$\rangle$ that usually have extremely high occurrences in the text, yet they seem to provide little information with the semantic relatedness of other words (e.g., "success","goal" and so on). One option to overcome this is the removal of such *stop words* from the basis elements (Lison and Kutuzov, 2017).

What if we have an extremely dense and large matrix? We wish to weight more on highly frequent context words that are informative to the target words, and less on the words that are ubiquitous. In other words, we want the semantically related words to have higher correlation and those with no semantic relatedness to have lower values (Terra and Clarke, 2004).

In following section, we introduce pointwise mutual information (PPMI), a weighting algorithm that allows to eliminate word pairs that were falsely correlated from the matrix. In our experiment, we use PPMI.

Positive Pointwise Mutual Information

Pointwise Mutual Information (PMI, Fano and Hawkins, 1961) is a measure of how often two events x and y occur together, compared with two events occurring independently. Given this definition, we can check if a target word and a context word co-occur more than if they were independent. The PMI between a target word w and a context word c is defined as following:

$$\text{PMI}(w,c) = \log_2 \frac{P(w,c)}{P(w)P(c)} \tag{2.1}$$

where $P(w)$ is the ratio of the number of times the word w appear in any contexts and the number of times each word and context in the text appears. $P(c)$ is the ratio between the number of times any words appear in context word of c in the text and the number of times each word and context in the text appears. $P(w,c)$ is the ratio between the number of occurrence w appears in context of c in the text and the total number of words and its contexts appeared in the text.

The $\text{PMI}(w,c)$ allows us to quantify an estimate of how much more the two words co-occur in a window than we expect by pure chance. In the nominator, we compute the probability of how often we see two words w and c together by using maximum likelihood estimates (MLE). MLE estimates the probability of some word x by normalizing the number of observations for x, c_x by the total number of

word tokens N:

$$\mathrm{P}(x) = \frac{c_x}{N} \tag{2.2}$$

Using the example of Jurafsky and Martin (2020), let us assume that we have a co-occurrence matrix F with W rows of target words and C columns of contexts (basis elements), we can get the count of word w_i and c_i co-occurring by accessing f_{ij} cell. Applying this to MLE, we can get $P(w_i, c_i)$ by:

$$\mathrm{P}(w_i, c_j) = \frac{f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}} \tag{2.3}$$

The denominator is the multiplication of the individual distribution, the probability of w and the probability of c, telling us that how often we would expect for the independently occurring words w and c to occur together. Knowing that w appears in the text might also tell us something about the likelihood of c being present, and vice versa. By taking the ratio of these two, we can get an estimate of how much more the two words actually co-occur together than we expect by chance.

The value of $\mathrm{PMI}(w,c)$ falls in a range of negative to positive infinity. If PMI is positive, then (w,c) pair is more likely to occur together since $\frac{P(w,c)}{P(w)P(c)} > 1$ and thus $P(w,c) > P(w)P(c)$, implying that w and c occur mutually more than individually. On the other hand, a negative PMI value means that the two words are co-occurring less often than both of w and c or one of them occurring individually. Its negative value tend to be unreliable since it is unlikely to get many co-occurrences of a word pair in a limited size of text, or otherwise it shows uninformative co-occurrences, for example, 'the' and 'book'(where word 'the' is extremely used). Thus the suggested solution for this problem is to use Positive PMI (PPMI, Niwa and Nitta, 1994). PPMI replaces all negative PMI values with 0:

$$\mathrm{PPMI}(w,c) = \max(\log_2 \frac{P(w,c)}{P(w)P(c)}, 0) \tag{2.4}$$

PMI is biased towards infrequent events. There are various ways to correct this bias empirically. One of them is to give rare words slightly higher probabilities (Jurafsky and Martin, 2020). A slight modification to the computation of $P(c)$ to $P_\alpha(c)$ solves the problem:

$$\mathrm{PPMI}_\alpha(w,c) = \max(\log_2 \frac{P(w,c)}{P(w)P_\alpha(c)}, 0) \tag{2.5}$$

where

$$P_\alpha(c) = \frac{count(c)^\alpha}{\sum_c count(c)^\alpha} \tag{2.6}$$

By raising the probability of the context words to the power of α (setting α to 0.75 has been found to be effective by Levy, Goldberg, and Dagan, 2015), the probability assigned to rare context words increases, and thus lowers their PMI scores.

Once the vector in semantic space is weighted, we can compute the similarity, distance or divergence between two words by using various similarity functions.

2.1.3 Distance Measures

Word pairs that have the highest similarity values (closest distance) computed by any distance measures are assumed to be semantically related. However, which measures are used is important as they can produce different performances during the evaluation phase. For our experiment, we use cosine similarity.

Cosine Similarity

The dot product of two vectors ($\vec{x}$, $\vec{y}$) is normalized by the division by the lengths of each of the two vectors. This is equal to the cosine of the angle between two vectors. Since the co-occurrence counts are non-negative, the range of the cosine for these vectors are in the range 0 to 1 with 0 being the lowest (the least similar) and 1 being the highest (the most similar).

$$sim_{cos}(\vec{x},\vec{y}) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \tag{2.7}$$

2.1.4 Syntax-based Models

Traditional word-based co-occurrence models build their vector space by only considering a window of co-occurring words surrounding the target word. Researchers (Padó and Lapata, 2007; Heylen et al., 2008) use these methods for the comparison to their own models. However, since our goal is to discover the effect of the integration of different linguistic information, we will not be focusing on this approach.

The abstract definition of context becomes sophisticated in the syntax-based models. The intuition of the syntax-based model is that we might be able to construct a semantically-enriched word vector space model that captures different semantic relations by incorporating information about the syntactic relationship between a target word and other words. Padó and Lapata (2007), Peirsman, Heylen, and Speelman (2007), and Heylen et al. (2008), each only consider the context words that satisfy a specific syntactic dependency relation to the target word.

Sentence:

He ate the cheese sandwich

Target words: ⟨*he, ate, cheese, sandwich*⟩
Basis elements: ⟨*(subj, he), (root, ate), (det, the), (mod, cheese), (obj, sandwich)*⟩

	(subj, he)	(root, ate)	(det, the)	(mod, cheese)	(obj, sandwich)
he	0	0	0	0	0
ate	1	0	0	0	1
cheese	0	0	0	0	0
sandwich	0	0	1	1	0

Figure 1: A simple example of Lin's (Lin, 1998) syntax-based semantic space.

Figure 1 shows an example of a matrix built by counting the co-occurrence of the syntactic relationship between words such as subject-verb and other relations. We can represent syntactic relations by tuple (r, w) where r is a relation type of a word w to a target word t. In this example, we use the basis vectors' term represented as (r, w). All the word-grammatical relation pairs in the example sentence constitute the basis vectors. We see a count of 1 in the cell of a row with a target word *ate* and a column of basis element *(subj, he)*, since in the example sentence, *he* is the subject of *ate*. Note that additionally, we can perform lemmatization. For example, the direct object of *ate* will correspond to the same basis vector of the direct object of *eat*. The idea is that by considering only the specific syntactic dependency relations, vector space model can be helpful to capture meaning of the target word.

Choices of what kind of syntactic relation one should use varies. Hagiwara, Ogawa, and Toyama (2008) have used indirect dependency in addition to normal direct dependency and shown the effectiveness in the acquisition of synonyms. In the study by Heylen et al. (2008), they consider eight syntactic relations (e.g., subject of verb, direct object of verb and modified by adjective). They find that their dependency model found more synonyms for high-frequency nouns and nouns that share semantic features of: object, event, property, situation, group, part, utterance, substance, location and thought. Padó and Lapata (2007) propose the use of *dependency paths* as contexts to build their vector space model. Given the dependency parse of the sentence, they define the context feature as *anchored* paths where the dependency starts at a particular target word. They only consider paths that have a maximum window size of k, that is, the absolute difference between the positions of the anchor (target) word and the context word with syntactic relation is at most k. They also discussed that some relations such as subjects and objects are more semantically informative than others. Thus they also constrained the context to be only a set of anchor paths with certain dependency relations. To quantify syntactic co-occurrence, they

defined a path value function where the function takes into account the obliqueness hierarchy of grammatical relations. However there is frequency bias where words occurrence is not distributed evenly. To avoid words wrongly considered as similar due to their similar frequency, Padó and Lapata (2007) use a lexical association function to remove those randomly co-occurring contexts.

Clark (2015) raises a potential problem that using the dependency relations can result in data sparsity due to considering only the refined notions of the context. To overcome this, Heylen et al. (2008) simply remove the frequency cut off, to include all the relations that appears, whereas Padó and Lapata (2007) define a basis mapping function to map a feature (r, w) to just a word w as their final basis element.

2.2 Semantic Relations

In this section we explain different semantic properties that VSM can be applied to extract.

Syntagmatic and paradigmatic relations distinguish two kinds of linguistic phenomena. Paradigmatic relation indicates a process of lexical selection whereas syntagmatic relations are mainly associated with co-occurrences of lexical units (Chiu and Lu, 2015).

2.2.1 Paradigmatic Relation

Words that are paradigmatically related are lexical units that are connected to each other by semantic relations and possibly, but not necessarily, by morphological ones (Polguère, 2016). To illustrate, we can think of relations between the base lexemes and derived lexemes such as $TEACH \iff TEACHER$ or $HOPE \iff HOPELESS$. Moreover, synonyms or quasi-synonyms like $UNDERSTAND \iff COMPREHEND$, antonyms like $ACCEPT \iff REFUSE$ and hyponyms/hyperonyms like $DRINK_{(N)} \iff TEA$ also constitute pradigmatic relations.

2.2.2 Syntagmatic Relation

Syntagmatically related words are ones that are likely to co-occur in the same text region. To illustrate: in English one *flies into* RAGE , but in French, one *puts themselves into* RAGE (*se mettre en* COLÈRE and in Russian, one *falls into* RAGE (*vpadat* RAŽ) (Mel'čuk, 2016). These phraseological expressions are called collocations and are gaining more and more importance in the description of linguistic phenomena.

2.2.3 Lexical Functions

In Mel'čukian Explanatory Combinatorial Dictionaries, the lexical functions have a crucial role in the lexicographic definition of a lexical unit. Defining the nature of relation between paradigms and col-

locations, lexical functions demonstrate by what other variety of the LU in question could be replaced (the paradigmatic aspect) and also with what other LUs it is likely to co-occur (the syntagmatic aspect). When called with a lexical unit L as an argument, the function f evaluates to L', this is notated as $f(L) = L'$. For instance, if the paradigmatic lexical function S_1 indicates 'someone who does...', then S_1(CRIME) gives CRIMINAL and analogically S_1(LECTURE) evaluates to LECTURER (Mel'čuk, 2016). The second kind of lexical functions constitute a syntagmatic operation. For instance, $Oper_1$ being the syntagmatic lexical function that has the meaning 'do...' then $Oper_1$(CRIME) gives us COMMIT and similarly $Oper_1$(LECTURE) evaluates to DELIVER since this word is the one that collocates with the word 'lecture' in this sense.

Since lexical functions denote a strong relation between lexical units, they take an important part in describing the relations between the word pairs that the VSM's catch. We cannot cover many lexical functions in this report but we will be following (Mel'cuk, 1996b) as a guideline for the detection of the lexical functions in our results.

2.3 Evaluation

Once we have VSMs, we then need to evaluate the quality of them. Here we discuss qualitative evaluation and quantitative evaluation methods.

2.3.1 Qualitative

Qualitative evaluation allows a deeper look into a program outcome. One can gain in-depth understand of "why" and "how" output is produced from a program by direct observation. One method of inspecting the quality of VSMs can be done by selection a set of words used to build VSMs and compute each word relation to others by using any distance measures. Levy and Goldberg (2014) evaluated their word embeddings by manually inspecting 5 most similar words and reasoning on its result. Pierrejean and Tanguy (2018) evaluated words with its computed neighbor words from distributional models and used the mean variation score with the standard deviation span between their default model and other variation model.

2.3.2 Quantitative

Quantitative evaluation methods involve the comparison between the manually collected gold data (e.g., thesauri) and the semantic space on specific intermediate sub-tasks (such as analogy completion and semantic similarity check). One downside of this evaluation method is that the automatically extracted VSM might capture correct semantic relations for some target word that are not listed in the manually created golden data. Also, the reliability of the gold data can be questionable. A common measure for

this evaluation is **precision at rank** *k*, that is a proportion of recommended words in the top-*k* set for a target word that are relevant. A total score is calculated by the sum of each target word's total number of how many predicted synonym words actually match the words in the "gold standard" thesaurus, and then average it by the number of target words.

For our experiment, qualitative approach is taken to evaluate our VSMs (Section 3.4).

3 Building Dependency-Based Context Word Embedding

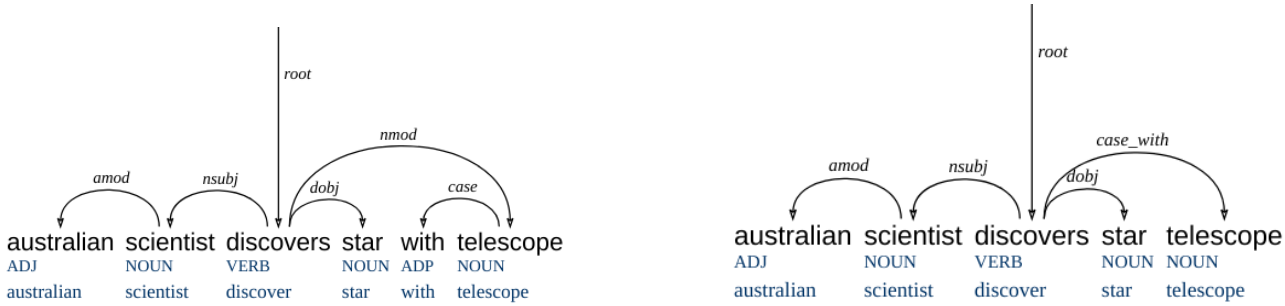
3.1 MANGOES software

3.1.1 Vanilla MANGOES

For our context similarity experiments, we used the MANGOES software [1] developed by magnet team in INRIA Lille Nord Europe. MANGOES is a toolbox for constructing and evaluating word vector representations. Implemented in Python3, MANGOES accepts different annotated text formats such as BROWN, CoNLL, and CoNLL-U. It also allows various representations used for dependency annotation such as Stanford Dependencies, Universal Dependencies, or a customized parser.

We can specify what vocabularies to be employed as target vocabulary and context vocabulary by applying different filters. Nevertheless, they only accept vocabulary to be wordform or a combination of wordform accompanied by its lemma and POS.

MANGOES has an option to build a co-occurrence matrix with window-based context and dependency-based context. Our research interest aligns with the latter, which defines contextual information based on the syntactic relations in which each word participates. Their implementation of dependency-based context is based on Levy and Goldberg (2014). Following their illustration, we will use the example sentence *"Australian scientist discovers star with telescope"* to describe how some parameters operate. The parameters that are available in dependency-based context are listed in the following:



(a) Dependency annotation

(b) Preposition collapsed

Figure 5: Example of a visualized dependency relations

[1] link: https://gitlab.inria.fr/magnet/mangoes/-/tree/master

- **collapse** : (bool.) If true, relations that include a preposition (in case of Universal Dependencies, it is *case* relation) are "collapsed" prior to context extraction, by directly connecting the head and the object of the preposition, and subsuming the preposition itself into the dependency label. To illustrate, in Figure 5a, there is a preposition relation between *with* and *telescope*. We get *with*'s head, which is *telescope* and directly connect it to *telescope*'s head, *discovers*. It replaces the dependency relation as *nmod* to `(preposition_relation)_(preposition_word)`, in this case becomes to *case_with*.

- **entity** : (tuple) One can specify attribute(s) to be considered. Thus, one can distinguish a word, where bark as NOUN and bark as VERB.

- **labels** : (bool.) Includes dependency relation, which results in more fine grained context words. For example, a word might share same lemma and POS, but different dependency relation will be counted differently.

WORD	CONTEXTS DIRECTED	CONTEXTS UNDIRECTED
australian		scientist/amod, discovers/amod+nsubj
scientist	australian/amod	australian/amod, discovers/nsubj, star/nsubj+dobj, telescope/nsubj+nmod
discovers	scientist/nsubj, australian/nsubj+amod, star/dobj, telescope/nmod, with/nmod+case	scientist/nsubj, australian/nsubj+amod, star/dobj, telescope/nmod, with/nmod+case
star		discovers/dobj, telescope/dobj+nmod, scientist/dobj+nsubj
with		telescope/case, discovers/case+nmod
telescope	with/case	with/case, discovers/nmod, star/nmod+dobj, scientist/nmod+nsubj

Table 3.1: Example of contexts for an example sentence with depth set to 2

- **depth** : (int.) This is similar to the notion of *path* introduced by Padó and Lapata, 2007. Words in a sentence not connected by direct dependency relation may hold relationship as well. Thus one can define how far a target word wants to include as its context using dependency connection. It can be considered as a similar idea of selecting window size used in a window-based context. In Table 3.1, the column CONTEXTS DIRECTED shows the example of what we get as contexts with depth set to 2. The setting of depth to k selects all the paths starting from target word w to the

linearly directed context word c, where the absolute difference between the two positions is at most the size k. For instance, from Figure 5a, *discovers* has direct dependency with *scientist* and so does *scientist* and *australian*. Thus we can say that *discovers* has indirect dependency with *australian* and satisfies our depth requirement that it's at most 2 steps away from *discovers*.

	(australian,ADJ)	(scientist,NOUN)	(discovers,VERB)	(star, NOUN)	(with, ADP)	(telescope,NOUN)
australian	0	1	0.5	0	0	0
scientist	1	0	1	0.5	0	0.5
discovers	0.5	1	0	1	0.5	1
star	0	0.5	1	0	0	0.5
with	0	0	0.5	0	0	1
telescope	0	0.5	1	0.5	1	0

	(australian,ADJ)	(scientist,NOUN)	(discovers,VERB)	(star, NOUN)	(with, ADP)	(telescope,NOUN)
australian	0	1	3	0	0	0
scientist	1	0	3	3	0	3
discovers	3	3	0	1	2	2
star	0	3	1	0	0	2
with	0	0	2	0	0	1
telescope	0	3	2	2	1	0

Table 3.2: Both tables are weighted matrix with depth set to 2 and path considered as undirected (paths can be found in Table 3.1 CONTEXTS UNDIRECTED). **Top** applies length-based weighting. **Bottom** is one that scores weight value with provided weight scheme {nsubj : 3, nmod : 2} and others are map to 1.

3.1.2 Extended Features

Still, MANGOES does not include some of the settings we needed to use to build the VSMs that are crucial to our experiments. Therefore, we implemented additional features in order to suit our needs. In the original implementation of MANGOES (vanilla MANGOES), we added the parameter to specify the maximum sentence length to be used. This upgrade was required since our raw text data contained sentences that were not tokenized correctly and resulted in peculiarly long sentences when they were parsed.

Furthermore, we expanded the functionality of the MANGOES dependency-based context. Considering the suggestions made by Padó and Lapata (2007), we added the following features:

- **depth** : (int.) We modified the definition of depth. As Padó and Lapata (2007) pointed out, confining ourselves to only directed paths (preserving the hierarchical head and modifier relation) may limit informative contexts where it could fail to capture, for instance, the relationship between the subject and the object of a predicate (e.g., *scientist* and *star* in Figure 5a). Hence we will be favoring the undirected dependency relations. We use a depth-first search algorithm to compute all the possible paths taken from the target word w. As shown in Table 3.1 CONTEXTS UNDIRECTED column, compared to CONTEXTS DIRECTED, we have more length of 2 paths.

- **directed**: (bool.) This is a new parameter we added. One can decide to restrict the path by only considering directed or undirected dependency path. Vanilla MANGOES considered dependency relation as directed graph. As we described in extended **depth** parameter, we should avoid limiting the context choices too severely.

- **deprel keep**: (list) Our research specifically targets syntagmatic relations, mainly collocations. For instance English collocation can be categorized into roughly four pairs of POS (e.g., Noun-Verb, Adjective-Noun, Adverb-Verb. Noun-Noun) thus we can only target certain dependency relations to be considered as context. According to the dependency relationship between the base and the collocate that is targeted, we can pass a list of dependency relationships to the method to only focus on those.

- **weight**: (bool.) Padó and Lapata (2007) introduce varying relative importance (e.g., value) on different paths. Traditional VSMs such as the co-occurrence matrix gives equal weight (e.g., 1) to all paths. Putting weight on more or less certain paths to others can provide more flexibility for incorporating linguistic information into the VSM. For this parameter, we simply apply values to paths by taking fraction of path length ($\left|\frac{1}{\text{path length}}\right|$). Thus if a path has a length of 2, then its assigned value is $1/2 = 0.5$

- **weight scheme**: (dict.) In addition to **weight**, we can also specify certain dependency relations to rank paths. For example, Padó and Lapata (2007) suggests the following scheme:

$$v(c,d) = \max\left(\begin{cases} 5, & \text{if } subj \in d \\ 4, & \text{if } obj \in d \\ 3, & \text{if } obl \in d \\ 2, & \text{if } gen \in d \\ 1, & \text{otherwise} \end{cases}\right) \tag{3.1}$$

  where v is the path value function and c is a context word, and d is its dependency relation contained in the dependency path to a target word w. It takes the highest value contained in the path. Thus applying different weights can weigh their respective contributions to VSM construction. Table 3.2 illustrates the use of weight and with weight scheme.

As our contribution, 878 lines of code were added and 782 lines of code were deleted in MANGOES.

Dependency-based context provides the flexibility of testing and combining different parameter settings. MANGOES has an option to apply several weighting functions to customize the built matrix. They are Joint Probabilities, ConditionalProbabilities, PMI, PPMI, ShiftedPPMI, TFIDF. It also has 22 choices of distance measures (e.g., cosine, euclidean, dice, Jaccard, yule) that we can use to compute similarities between two words in VSMs.

3.2 Corpora

Since our task is language-agnostic, we targeted French and English languages. We first collected texts from Wikipedia Monolingual Corpora [2], then, since the original text was in XML format, thus we only extracted the body of the text we required. Once we collect only relevant texts, we split the text into sentences. Corpus cleaning was performed using command-line utility. We needed our text to be formatted with CoNLL-U format. Each line represents a single word with ten different tab-separated fields since it is one of the supported formats as input of MANGOES. They include: word index (ID), word form (FORM), lemma (LEMMA), universal part-of-speech tag (UPOS), language-specific part-of-speech tag (XPOS), list of morphological features from the universal feature inventory (FEATS), head of the current word (HEAD), universal dependency relation to the HEAD (DEPREL), enhanced dependency graph in the form of a list of head-deprel pairs (DEPS) and then, any other annotation (MISC). Missing fields are replaced with an underscore.

In our case, the information on UPOS/XPOS, DEPREL and HEAD are our interests in the experiment. To parse the raw sentences, we chose to use the Stanford CoreNLP parser [3]. The parsing was done using Grid'5000 [4], a large-scale and flexible testbed to perform research experiments for researchers. Due to the amount of the text, it took several weeks to parse the text using resources of Grid'5000. We had to reduce the text data size later at the experiments phase due to the memory issue. The output of the Stanford CoreNLP parser produced XPOS instead of UPOS for English as we only could find the POS tagging that uses XPOS. Our corpus is described in Figure 6.

We then present the statistics of POS that our corpus contains in Figure 7 for English and Figure 8 for French (page 16). For English, the rest of the POS (JJR, JJS, AFX, UH, RBR, . , LS, $, MD, ,, PRP, DT, CC, RP, :, RBS, HYPH, PDT, PRP$, -RRB-, WRB, ‘, WP, -LRB-, ”, TO, WDT, POS, WP$) have the count of less than 1000.

We also demonstrate the dependency relation counts in Figure 9 for English and Figure 10 for the French corpus (page 17). Remaining dependency relations (nmod:range, obj:agent, discourse, iobj:agent, dislocated, reparandum, vocative, goeswith, csubj:pass, advcl:cleft, orphan, flat:foreign,

[2] https://linguatools.org/tools/corpora/wikipedia-monolingual-corpora/

[3] https://stanfordnlp.github.io/CoreNLP/

[4] https://www.grid5000.fr/w/Grid5000:Home

Language	English	French
Number of Sentences	19182562	18916629
Number of Tokens	518854512	494582365
Number of Unique Tokens	4793040	3949618
Number of Tokens without stop-words	4787871	3941659

Figure 6: Corpus description

compound, parataxis, ccomp, expl:pass, expl, iobj, dep, aux:caus, csubj, nsubj:caus) are counted less than two million times.

For the French corpus, remaining dependency relations (compound, obl:npmod, expl, csubj, iobj, cc, det:predet, orphan, csubj:pass, discourse, goeswith) are counted less than one million times. We use the corpus that we built to obtain our VSMs and we explain the pre-processing that we have applied to it in the next section. There are, however, some limitation in our corpus that could influence our experiment results: (1) some text were not ended with end of sentence punctuation, thus they might failed to parse into correct sentences, and (2) wrong lemmatizations are produced from some words in text. For example, one can still see the masculine and feminine forms of the same adjective or the different conjugations of the same verb in French. We will be observing more detailed examples in Section 3.4.2.

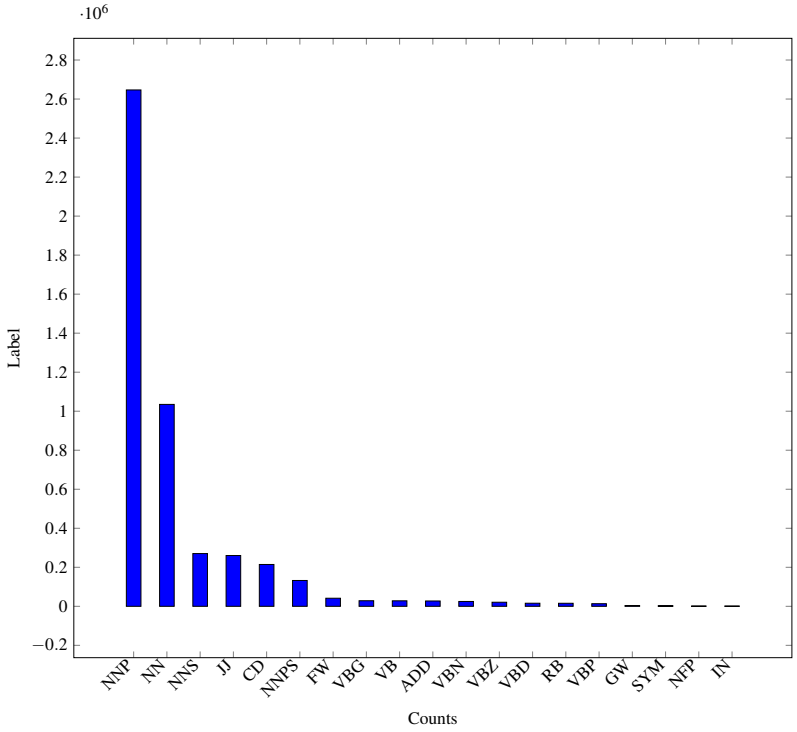


Figure 7: POS count for English corpus

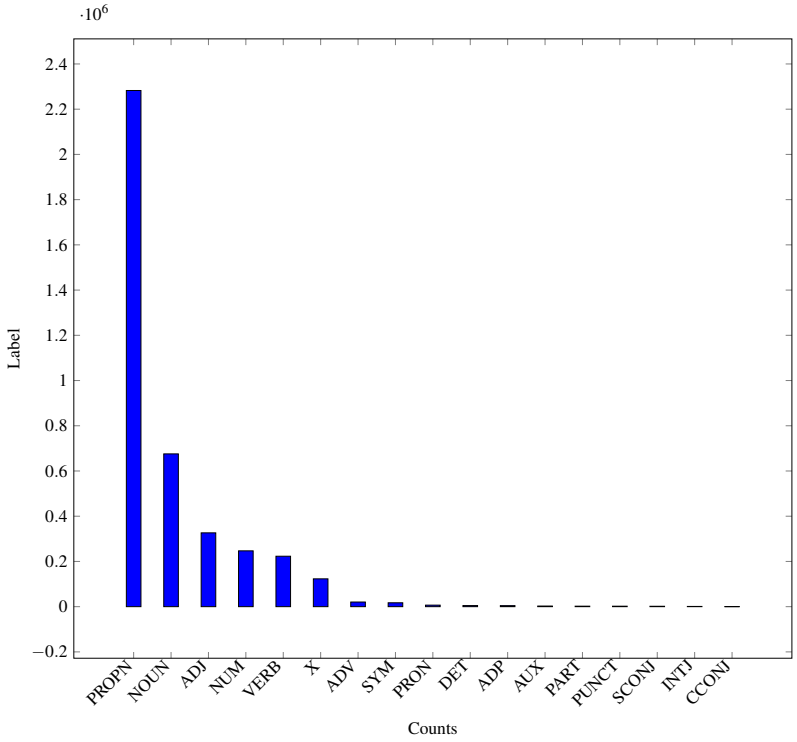


Figure 8: POS count for French corpus

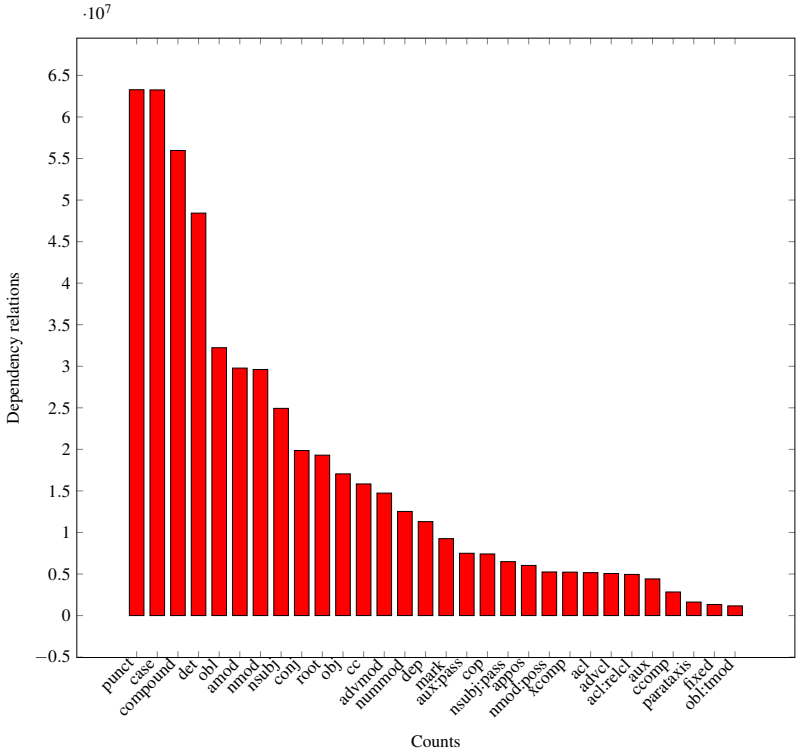


Figure 9: Dependency relation count for English corpus

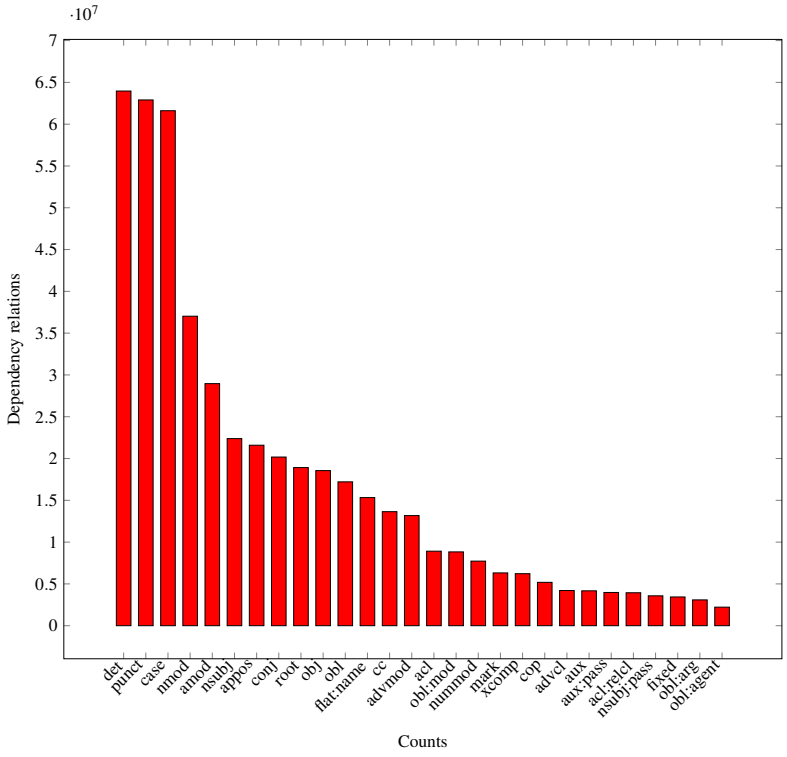


Figure 10: Dependency relation count for French corpus

3.3 Experiments

3.3.1 Parameter Settings

We first describe the base settings of the experiment. We discard sentences that are longer than 100. The reason of setting the maximal sentence length is because the sentence tokenization could not parse some of the sentences correctly during the corpus creation. We chose 100 to be the suitable number to proceed with our experiments due to the hardware memory issue.

We feed our co-occurrence matrix to weighting function, PPMI as explained in Section 2.1.2, and then apply singular value decomposition (SVD) afterwards. SVD is a well-known dimension reduction used to factorize the matrix produced by PPMI. We reduce the dimension of each target words to have 300. At the end the size of our embedding is $|\text{target vocabulary}| \times 300$. The main objective of the experiments is to explore whether syntactic features extracted from the text suffice to capture strong collocations.

In general, we take entity token in the form of (lemma, POS) and eliminate the wordform since the paradigms of the same lexeme is not relevant to the collocation. Thus if in a sentence uses words *has* and *had*, both words' lemma are *have* and so they will map to same basis element as well as context element. Furthermore, it keeps the matrix from becoming too large and sparse.

For the default dependency-based context, we enable collapsing in order to connect possible collocations connected by prepositions. To prevent from matrix to grow too large, we do not consider the uniqueness of the dependency relation but the uniqueness of the token (pair of word's lemma and its POS). We also set the depth to 1 and consider the dependency relation to be undirected.

When it comes to the default vocabulary settings, we skip stop-words (provided by NLTK package available in Python) from the vocabulary. However, we decided not to exclude certain words based on their frequency. For example, in the English corpus, we noticed from analyzing word frequency that the top 5% most frequent words consist of prepositions and symbols, but also includes some of the words we think should be included such as *make*, *take*, *hold* and others. As our knowledge, we think those light verbs are popularly used in collocation. We could think of using those words, for example, *hold classes*, *make a room*, *take a chance*, *take a bow* and so on. As we explained in Section 3.1, we can apply ranked dependency relations to give different weights in our VSMs. Thus, depending on the dependency path, the least frequent words can have an impact to VSMs.

We now explain in detail the choice of parameters used during the construction of VSMs. As our vocabulary setting, we have tried three different ways: applying POS filter to only target vocabulary, only context vocabulary, and both vocabulary. In order to cover four significant types of collocations, namely, Noun-Verb, Adjective-Noun, Adverb-Verb, Noun-Noun, for each option, we applied (ADV, ADJ, NOUN, VERB), (NOUN, VERB, ADJ), (NOUN, ADV, VERB) respectively. Combining the three POS filter options and three vocabulary options, then the two choices of depth yield eighteen

model instantiations.

We tested paths of length 1 and 2 as context for the depth setting of dependency context. The depth of 2 considers path length of at most two, and it is able to cover phenomena such as coordination, genitive constructions, noun compounds, and other sorts of modification.

We constructed three path value functions explained in Section 3.1:

- *base* assigns the value of 1 to all counted paths. It assumes that all paths are equally important.
- *length* assigns each path a value inversely proportional to its length. It discourages the weight to longer paths.
- *gram-rel* defines ranking paths according to its dependency relations. We defined following weight schema:
  - {pobj: 5, dobj: 5, iobj: 5, obj: 5, nsubj: 5, obl: 5} partially aligns with the Equation 3.1 where the obliqueness hierarchy of grammatical relations are taking into account.
  - {amod: 5} gives more weight to an adjectival modifier of a noun.
  - {advmod: 5, advcl: 5} gives high values to an adverbial modifier and a clause which modifies a verb or other predicate (adjective, etc.).
  - {nmod: 5, compound: 5} allows nominal modifier and relations for multiword expressions to account as important.
  - {advmod: 5} only allows an adverbial modifier relation to have a high value.

The combination is tested with each depth setting, which yields thirteen models.[5]

Different from path value functions, we also experimented with restricted dependency relations. We constrained our context words to contain only selected dependency relations {pobj, dobj, iobj, obj, nsubj, obl} in their paths. The choice of dependency relation is partially based on Equation 3.1. In total, we have experimented with thirty-three models.

3.3.2 Running the experiments

To train the VSMs we required huge computational resources, thus we used the computational cluster Grid'5000 during the project. For the depth 1 models, we utilized available hardwares with memory of 128 GiB which took around 4 hours per experiment. However, for the depth 2 models, they required more memory than depth 1 models and there was only one hardware resource, namely *graphite*, with 256 GiB and 4 cores available. Each experiment were done at night or weekends and took around 7 hours since the longer jobs were not allowed to run at weekdays for the *graphite*. Due to the limited resources and time, we could not further explore parameters such as combination of different parameters and more depth.

[5] Because the *length* with depth 1 and *base* with depth1 is equivalent.

3.4 Qualitative Analysis Evaluation

Although we have introduced possible evaluation methods in Section 2.3, one thing we lack in order to perform quantitative evaluation is the gold data. We built the VSMs targeting to acquire collocation and finding out about its lexical function encoded in the embeddings. Thus we need an extensive collocation dictionary that is ideally sorted by lexical functions. However, we were unable to find any publicly available dataset, and collecting such a dictionary could be a whole separate project. So instead, for our evaluation, we explore the variation between VSMs trained with only one/two varying parameter(s) by observing the distributional neighbors and showing how one parameter can impact a VSM.

We manually inspect ten nouns with their most similar words with different POS to a given set of target words for English (Table 3.3 on page 21) and French (Tble 3.4 on page 24). Some of the target words are selected based on Mel'cuk, 1996b's examples (e.g., cry, rain, price) and others are randomly chosen. The reason we chose to only select different POS from its target word are because we want to capture more of syntagmatic relations instead of paradigmatic ones. Although there is no guarantee of selecting different POS producing mainly paradigmatic relations, however, based on our inspection of the top 5 most similar words in order regardless of the POS, we mainly saw its paradigmatic relations or same POS as results, even with our syntactic information embed VSMs.

We used cosine similarity (explained in Section 2.1.3) as distance measurement. Table 3.3 and Table 3.4 show the results of the top 5 similar words collected from five embeddings. The five embeddings are chosen by manually inspecting all the thirty-three embeddings with similar words from given target words and determining the models with more promising output than other models.

Our analysis, especially in terms of analyzing the syntagmatic relation between words into lexical functions are based on our assumption since there are no complete lists of words categorized by lexical functions (Mel'cuk, 1996a, Mel'cuk, 1996b).

3.4.1 Results of the English Embeddings

Models used in Table 3.3 are following:

- BASE: depth 1 + *base* path value function
- SVD2: depth 2 + *length* path value function
- SVD3: depth 2 + *gram-rel* path value function with weight scheme of {pobj: 5, dobj: 5, iobj: 5, obj: 5, nsubj: 5, obl: 5}.
- SVD4: depth 2 + dependency relation filter of {pobj, dobj, iobj, obj, nsubj, obl}.
- SVD5: depth 2 + POS filter of {ADV, ADJ, NOUN, VERB} to both target and context vocabulary

Interestingly, we have observed that the words chosen as similar to the target words are slightly improved with model depth set to 2. These 5 models are chosen to illustrate how distinguish settings (different weight value computation, dependency relation filter, POS filter) can impact results of SVDs.

Table 3.3: Target words and their 5 most similar words, as induced by different VSMs.

WORD	BASE	SVD2	SVD3	SVD4	SVD5
rain	breath, alight, muddier, melting, standstill	warm, hail, trickle, drowned, mist	hail, drowned, melting, snow, breeze	simmable, warm, rainy, hot, quiet	rainstorm, warm,hail, snow, breeze
cry	hear, laugh, remember, ... , hate	shout, hear, scream, loudly, laugh	horrible, awful, loudly, shout, silently	bad, beloved, evil, shout, hear	hear, recall, shout, scream, loudly
price	non-monopoly, supra-competitive, re-roll, minimum, reasonable	worth, pay, net, cost, exceed	worth, net, cost, pay, financial	million, billion, chronically, multi-day, cost	worth, net, pay, profitable, exceed
proposal	agree, propose, decide, urge, recommend	propose, decide, agree, approve, reject	reject, agree, decide, propose, approve	propose, reject, approve, submit, upon	propose, approve, reject, agree, recommend

Continued on next page

Table 3.3 – continued from previous page

WORD	Base setting	SVD2	SVD3	SVD4	SVD5
support	provide, oppose, cladd, gutter, peg	provide, oppose, political, concrete, externally	political, military, future, seek, oppose	oppose, seek, gain, provide, declare	provide, lead, brace, concrete, flush
illness	suffer, chronic, subacute, post-partum, atopic	suffer, afflict, traumatic, fatal, severe	ill, suffer, afflict, fatal, traumatic	ill, transgendered, sententiae, die, diagnose	suffer, afflict, ill, chronic, rheumatic
attention	give, bring, show, pretend, warn	scheme, intent, aware, meet, bring	vain, distressed, desperately, scheme, consummate	draw, bring, attract, show, turn	bring, draw, meet, frustrated, seduce
conference	organise, host, organize, sponsor, episcopal	organize, participate, sponsor, inaugural, organise	participate, organize, invite, select, sponsor	organize, select, invite, attend, sponsor	sponsor, participate, organise, invite, episicopal
deal	sign, join, pay, discuss, concern	sign, join, relate, address, concern	sign, announce, relate, address, discuss	sign, loan, discuss, concern, relate	sign, announce, agree, sign, discuss
pain	cough, afflict, debilitate, suffer, bruise	painful, chronic, acute, severe, traumatic	painful, cough, fatal, traumatic, chronic	sick, ill, insane, painful, sudden	painful, cough, chronic, sweat, sore

Surprisingly, for the target word *rain*, many of the similar words describe the particular *rain* situation such as *mist*, *warm*, *muddily*, and *quiet* but no strong collocation is observed. In contrast,co-hyponyms of *rain* such as *hail*, *snow*,and *rainstorm* constitute a successful list of paradigmatically related list of similar words.

When it comes to the target word *cry*, compared to the base setting, the other models provide more reasonable lists of similar words. The models with the depth 2 setting propose the quasi-synonym of the target word which is *shout* and SVD2 and SVD5 also got the quasi-synonym *scream*. The word *loudly* appears in SVD2 and SVD5 which is a strong collocation to the verbalized form, *to cry* of target word, being the result of the lexical function *Magn*; usually one says *to cry loudly* to intensify the act of crying. However, we considered *cry* as a noun and not a verb, so this means there may be a disagreement between homophones on the level of POS. Moreover, the model SVD3 strong adjectival collocates like *horrible* and *awful* which can be explained through $AntiBonMagn(cry) = horrible,\ awful$.

The varied settings resulted the target word *price* to yield very different similar words from the base setting, which did not perform very well by proposing rare words like *non-monopoly* and *supracompetitive*. On the other hand, SVD2, SVD3 and SVD5 captured the $Oper_1$ collocate of the target word which is *pay*. Other than that, we observe many words related to finance such as *financial*, *profitable* etc. but no strong or direct relatedness is detected.

For the *illness*, the word *ill* (S_1) that appears in SVD3, SVD4 and SVD5, is a paradigmatically related word to the target word. However except the model SVD4, the word *suffer* appears in all the models which is the outcome of the syntagmatic function $Oper_1$ applied to *illness* as one can say *suffer from [ART] illness*.

Different SVDs contain somewhat similar verbial words when it comes to *attention*. We observe that the models SVD1, SVD2, SVD4 and SVD5 has *bring [X to ART]* $\sim$, SVD4 and SVD5 also contain *draw [ART]* $\sim$, SVD4, which seem to perform well on this target word, also propose *attract [X's]* $\sim$; and all can be categorized by lexical functions $Caus_1Oper_2$, $CausOper_1$ and $Caus_2Func_2$ respectively.

The target word *conference* as well showed important verbial similar word among tested SVDs. For instance, one can say *participate [ART]* $\sim$. In terms of categorizing into lexical functions, *participate* and *attend* belong to $Oper2$; whereas *host* and *organize* can be categorized into $Caus_1Func_0$.

The *pain* example illustrate interesting results. Compared to the BASE, the other models describe the type of pain such as *acute pain* and *chronic pain* which can be explained by the functions *Magn* and $Magn_{temp}$ successively. In fact, for the SVD2 result, it contains *severe*, which intensify the meaning of pain, can also be categorized as *Magn*.

We can conclude that the similarities collected from our constructed VSMs do not always produce homogeneous relations. Most of our example target words tend to be used together with similar words because such combinations are pretty specific, and the chance of occurring both the base and collocate is high. We listed more words with their similar words for English in Appendix A.1.

We noticed that overall many target words result in similar sets across the different setups. However, other target words show varied selected similar words with different VSMs compare to BASE model.

3.4.2 Results of the French Embeddings

As we mentioned in Section 3.2, we applied the models on our French corpus as well. Models used in Table 3.4 are following:

- BASE: depth 1 + *base* path value function
- SVD2: depth 2 + *length* path value function
- SVD3: depth 2 + POS filter of {ADJ, NOUN, VERB} to both target and context vocabulary .
- SVD4: depth 2 + *gram-rel* path value function with weight scheme of {nsubj: 5, csubj: 5, dobj: 4, iobj: 4, obl: 3}.
- SVD5: depth 2 + POS filter of {ADJ, NOUN, VERB} to both target and context vocabulary + weight scheme of {nsubj: 5, csubj: 5, dobj: 4, iobj: 4, obl: 3}.

We chose these models to include in the Table 3.4 because firstly, we intent to demonstrate change between the depth 1 and the depth 2 settings of MANGOES that we explained in Section 3.1.1, which, in the case of the French corpus, is quite drastic. SVD3 and SVD4 yielded the best results out of the filters and the features that we used, and thus in the SVD5 we decided to merge SVD3 and SVD5 to see how they work together.

Table 3.4: Target words and their 5 most similar words, as induced by different VSMs.

WORD	BASE	SVD2	SVD3	SVD4	SVD5
pluie	leau, lhumidité, brûlant, dair, daérolithes	lentement, leau, souffler, tombant, froide	pleuvoir, souffle, lhiver, prémunie, réfractent	lentement, souffler, tombant, leau, contin-uellement	pleuvoir, lhiver, souffler, prémunie, gelé
voyage	ûllah, voyager, katius-cas, part, titgharrab	voyager, partir, visiter, raconter, arriver	voyager, raconter, partir, arrivé, relate	voyager, visiter, venir, arrivé, partir	voyager, part, raconter, partir, arrivé

Continued on next page

Table 3.4 – continued from previous page

WORD	Base setting	SVD2	SVD3	SVD4	SVD5
admiration	recessional, lexten-sibilité, kurstaki, 860-863,	admirer, intime, lamitié, desprit, apprécier	éprouver, sincère, voue, lamitié, desprit	admirer, apprécier, admirer, intime, lamitié	admirer, lamitié, éprouver, sincère, exprimer
proposition	lidée, de-mandant, khorochko, no-o-war-r, linitiativ	voter, ac-cepter, re-fusée, re-jeter, re-jeté	rejeter, accepter, vote, refuser, voter	voter, refuser, rejeter, accepter, préciser	rejeter, refuser, accepter, voter, adopter
examen	lexamen, tri, traite-ment lintérêt, posteriori	préalable, examiné, médical, lexamen, examiner	examan, adque, cyto-bactériolo-gique, solenne, préalable	préalable, examiner, lexamen, médical, examiner	examan, adque, cyto-bactériologique, solenne, trans-crânien
argument	largument, priori, évidem-ment, ceci, suppose	saurait, claire-ment, juste-ment, évident, justifier	évident, logique, savoir, justifier, contredire	évident, juste-ment, claire-ment, saurait, logique	largument, logique, évident, con-tredire, savoir
attention	sinon, au-tant, soi, lintérêt, évidem-ment	apporter, évidem-ment, constater, certes, oublier	apporter, constater, rappeler, oublier, compren-dre	évidem-ment, apporter, certes, constater, justement	apporter, évidem-ment, constater, oublier, justement

Continued on next page

Table 3.4 – continued from previous page

WORD	Base setting	SVD2	SVD3	SVD4	SVD5
conférence	congrès, lassemblée, lunion, anfrangen, ...	organisé, organisée, organiser, nations, présider	uni, organiser, organisé, européenne, réunire	congrès, organisé, organiser, nations, organisée	unies, européenne, réunir, international, présider
accord	laccord, pacte, conclure, prévoire, lacte	conclure, prévoir, accepté, décidé, ...	signé, conclure, signer, négocié, négocier	conclure, prévoir, signer, décider, accepter	signé, conclure, négocier, signer, prévoyant
applaudissement	détonnement, soir-là, écoutant, criant, bruyament	applaudir, saluer, dentendre, bruyament, soir-là	applaudi, huer, saluer, applaudit, applaudir	bruyament, saluer, applaudi, soir-là, applaudir	applaudi, applaudir, bruyament, huer, saluer

We can already observe that the bad lemmatization of the French corpus that we mentioned in Section 3.2 has affected the most similar words that we retrieved for the target words. Table 3.4 shows that more than one form of the same lexeme often appear together in the same cell. For instance, *examiner* ('to examine') infinitive form appears along with *examiné* ('examined'), conjugated in past tense as a similar word to *examen* ('exam') on the model SVD2. It goes without saying that this situation has affected the context similarity task in French in a negative way since the paradigms belonging to the same lexeme occupy the position of other possible similar word candidates.

Moreover, we often observe words that are still attached to the *l* or *d* [6] and only the apostrophe is removed thanks to the punctuation removal. So, for example *l'homme* ('the man') remains as *lhomme* in many cases. However, for the sake of evaluation, we will ignore this and consider *lhomme* as *homme* and so on.

[6] in French, the definite articles *le* and *la* reduce as *l'* and the preposition *de* reduces as *d'* when followed by a word starting with a vowel or a silent *h*

As in most of the target words in French, the base setting yielded considerably feeble results and even dummy or foreign words such as *titgharrab* for *voyage* ('travel'), *kurstaki* for *admiration* ('admiration'), *khorochko* for *proposition* ('proposal') etc. Nevertheless it was still able to catch some interesting related words such as *humidité* ('humidity') whose relation to *pluie* ('rain') is somehow questionable but might be explained as S_{res}(*pluie [in location Y]*) = *humidité [in location Y]*. Furthermore, the base setting model managed to catch *pacte* ('agreement') as a neighbor to *accord* ('deal') which is its quasi-synonym.

For *pluie*, the models SVD2 and SVD4 proposed an important neighbor *tombant* ('falling'), whose correctly lemmatized form *tomber* ('to fall') is the collocate of *pluie* initiates it by the lexical function $Func_0$. The models SVD3 and and SVD5 catched simply the verbalized *pleuvoir* ('to rain'); and additionally, speaking of the verbalized form of *pluie*, the model SVD4 also proposed *continuellement* ('continuously') whose relation to *pleuvoir* could be justified by the lexical function $Magn_{temp}$.

When it comes to the target word *voyage*, all the models have the verbalized *voyager* ('to travel'). Furthermore, every model except the base setting manages top retrieve the $IncepOper_1$ collocation *partir* ('to leave/ to go on') since one says *partir en voyage* ('to go on a travel') in French.

The results for the target word *admiration* are interesting since the models came up with adjectival and verbal collocates for it. Every model except the base one has got the verbal paradigmatic relation to *admirer* ('to admire'). Additionally, SVD3 and SVD5 proposed *sincère* ('sincere') *Ver* collocate of the target word in question since it gives the meaning of 'genuine'. The same models were also able to catch *éprouver* which is related to the target word by the lexical function $Oper_1$.

For proposition, every model except the base setting managed to retrieve collocates such as *accepter* ('to accept') as in $Real_3$(*proposition*) = *accepter* and *rejeter* ('to reject') and *refuser* ('to refuse') as in $AntiReal_3$(*proposition*) = *rejeter, refuser*.

Although *passer* ('to pass/ to take') would have been satisfying to observe within the list of the most related words of *examen*, the only considerable similar word that is captured by the models is *examiner* ('to examine').

When it comes to the target word *argument*, the collocate *logique* ('logical') captured by the models SVD3, SVD4 and SVD5 is a good observation and the relation between the collocate and the base could be explained by the lexical function *Ver* since, by definition, an argument should make sense.

Every model except the base setting managed to get the $Oper_1$ collocate of the target word *attention*, which is *apporter* ('to bring'); in this case, *apporter [ART] attention* carrying the meaning 'to pay attention'.

Furthermore, the $Caus_1Func_0$ collocate for *conférence* ('conference') which is *organiser* ('to organize'), is captured by the models SVD2, SVD3 and SVD4 as well as some quasi-synonyms such as *congrès* ('congress') and *assemblée* ('gathering').

For the target word *accord*, the collocates that we can observe are *négocier* ('to negociate') and *signer* ('to sign'), which could be explained by the lexical functions $IncepOper_1$ and $Caus_1Func_0$

respectively.

And finally, when it comes to *applaudissement* ('applause'), the models with depth 2 setting retrieved the paradigmatic relation to *applaudir* ('to applause'). SVD3 also proposes *huer* ('to boo') which is the antonym of applause. On the other hand, SVD2, SVD4 and SVD5 captured the collocate *bruyamment* ('loudly') which is the *Magn* intensifier not of *applaudissement* but of *applaudir* since it is an adverb, which is still indirectly related to the target word. More French examples can be found in Appendix A.2.

From our results, we can say that it is not possible to target only one type of relations with our tested parameter settings. More specifically, some target words are able to capture only paradigmatic relations (see *proposal* in Table 3.3) whereas other words are mixed with syntagmatic and paradigmatic relations.

We would like to note that our VSMs have several limitations. Our VSMs are affected mainly by (1) the parsed quality of the text, (2) limited collocate of the collocation included in the text, and (3) only particular dependency relation of the collocation occurred. However, we believe our project is an essential step toward understanding how syntactic information can impact the quality of defined VSMs and why. We discuss in Section 4 about what could be possibly used from our project.

4 Conclusion and Discussion

4.1 Discussion on the result

From our manual analysis of top 5 similar words given target words, we found that with our experimented parameter settings, it is not possible to target only syntagmatic or paradigmatic relations for both English and French. It could also be word specific. Some target words seem to capture more words with paradigmatic relations. Thus we suggest that one needs to be careful using our dependency-based VSM settings especially if the tasks require strong capture of certain type of relations.

4.2 Conclusion on the realized work

To highlight our main contributions in this project:

- Prepared corpus from raw text utilizing command line and Stanford core NLP parser with grid5000.
- Extended several functionalities, especially dependency-based context parameters in existing software MANGOES.
- Validated the functionalities with unit-testing.
- Experimented with various parameter settings and built VSMs using grid5000.
- Manually analyzed the result with qualitative evaluation.

Our principal objective was to determine a noticeable difference in similar words computed from constructed VSMs by changing parameters relating to syntactic information. Although additional evaluation will be required to compare the different models, we can conclude that dependency-based VSM can generate different similarities of paradigmatic and syntagmatic relations.

4.3 Challenges and limitations

For the second part of the Supervised Project, we faced three complex challenges. We first faced difficulty obtaining quality text needed for our experiments. We had to create a corpus on our own since there were no publicly available datasets we needed. Processing data on our own also resulted in taking some time. Finding the way to get all the necessary annotations using Stanford core NLP

sometimes did not yield the output we wanted. However, we were able to obtain the dataset with our desired output to be used in our experiments. Due to the hardware constrain, we realized the amount of text was too big to fit the available hard disk. Thus we had to take a subset of the corpus. Additionally, one can add and use enhanced-dependency annotation, which fine-grain the dependency relation, such as making some of the implicit relations between words more explicit, etc. A more in-depth look into different kinds of dependency relations and understanding them can be notably helpful.

The majority of the time was spent familiarizing ourselves with MANGOES software, and we faced the difficulties of modifying the code that was written by others: fixing the bugs and writing the unit tests to ensure our implementation, especially the dependency-based context, to work correctly. Sometimes, the function we thought was doing X was instead of doing Y, and the lack of detailed comments has challenged our understanding of MANGOES.

We also struggled to run the experiments successfully since some of the time, jobs are never completed due to the limited memory and resources available to us. For example, some of the embeddings took more than 7 hours to be completed, and we ran thirty-three models.

4.4 Future Improvements

This report has noted that many parameter settings could be used to explore and observe different outcomes possibly seen in VSMs. We think the quality of the text is paramount. Thus, it is necessary to spend more time analyzing the text and ensuring rich collocation examples are included. However, handling sizeable textual data costs large memory. One can define a more fine-graded selection of target and context vocabularies.

MANGOES software can be extended more to accept various parameters. Also, the existing parameters can be tested in more depth. For example, in dependency-based context, we only tested with depth up to 2 and did not combine different settings. However, there are too many combinations one can experiment with, and it is more reasonable to experiment with only one or two different settings. Furthermore, coming up with more weighting schema for path value function can be investigated further. We did not explore all the similarity measures and weighting functions to note the difference in VSMs. This is because we were more interested in the impact of dependency-based context on VSMs.

As for the evaluation, if an available collocation dictionary dataset is categorized into corresponding lexical functions, we can perform a quantitative evaluation.

The idea of distributional hypothesis where nearly co-occur words usually have a high chance of having semantically related meanings seems intuitive; however, explaining how collocation occurs and describing each meaning is difficult in any language. While working on this project, we have exposed ourselves to different literature related to this research. Furthermore, by understanding what information impacts the quality of target features/relation encoded in VSM, we see how this research can further help with different NLP tasks.

A Appendix

A.1 Additional Target words and their 5 most similar words

Table A.1: Similar words computed from various VSMs for English

WORD	Base	SVD2	SVD3	SVD4	SVD5
smoker	smoke, spit, addict, dream, quip	smoke, addict, spit, lame, obese	lame, smoke, obese, bother, unhealthy	alcoholic, adult, adolescent, elderly, unemployed	smoke, addict, breastfeed, obese, underestimate
drunk	smoke, beggar, miserable, distracted, smell	smoke, unruly, victimize, scare, filthy	intoxicated, filthy, unruly, disguise, rowdy	begger, nowhere, runaway, blonde, terrifying	smoke, annoy, disguise, victimize, filthy
sleep	happen, wake, sweat, experience, alone	tear, wake, suffer, clean, experience	nervous, sick, awake, painful, wake	human, whatever, sick, little, unconscious	awake, spy, hallucinate, wait, wake

Continued on next page

Table A.1 – continued from previous page

WORD	Base setting	SVD2	SVD3	SVD4	SVD5
optimistic	seem, temper, understate, like, kind	quite, genuinely, seem, appreciate, optimism	optimism, genuinely, seem, appreciate, temper	disappoint, anticipate, mirror, applaud, criticise	genuinely, surprisingly, intensely, seem, profoundly
friend	tell, know, learn, befriend, old	meet, tell, portray, young, stay	younger, marry, tell, meet, say	old, know, ask, tell, married	tell, marry, learn, reveal, see
smell	soak, raw, boil, consume, fresh	poisonous, watery, consume, soak, unfiltered	poisonous, watery, harmless, cold, unpleasant	ingest, soak, sweet, consume, swallow	unpleasant, mask, mix, unfiltered, pop
condemn	heresy, treason, guilty, accusation, protest	condemnation, accusation, harshly, heresy, conspiracy	openly, unacceptable, harshly, condemnation, allegedly	tantamount, complicit, motivated, guilty, protest	punishment, condemnation, heresy, accusation, protest
rely	solely, suited, concerned, superior, acceptable	importantly, acceptable, solely, reliance, careful	importantly, regardless, solely, better, aside	limited, easier, reliant, acceptable, sufficient	solely, reliance, importantly, aside, reliant

Continued on next page

Table A.1 – continued from previous page

WORD	Base setting	SVD2	SVD3	SVD4	SVD5
laugh	crazy, nice, kind, thrill, dumb	crazy, awful, nice, smile, mind	crazy, nice, awful. annoying, youre	funny, crazy, youre, always, nice	crazy, mind, nice, tease, 're
accent	rhyme, uniquely, distinctly, fluent, distinctively	voiced, spoken, colloquial, tense, rhyme	tense, spoken, voiced, colloquial, phonetically	tense, verb, consonant, russian, greek	spoken, contrast, tone, pattern, pronounce
order	request, demand, refuse, force, allow	request, demand, force, refuse, send	immediately, upon, personally, secret, military	general, request, send, refuse, upon	allow, send, refuse, immediately, decide
resistance	suppress, prevent, resist, surge, damp	suppress, resist, prevent, rectify, minimise	suppress, resist, weak, strong, effectively	general, british, resist, particular, latter	resist, suppress, prevent, fight, minimise
promise	agree, ask, seek, want, demand	unwilling, accept, decide, demand, insist	unwilling, willing, wish, insist, compelled	right, ask, demand, accept, refuse	agree, willing, wish, unwilling, accept
complaint	cite, allege, report, complain, appeal	allege, deny, file, respond, report	allege, file, sue, unfair, improper	file, allege, deny, respond, report	allege, deny, file, dismiss, sue

Continued on next page

Table A.1 – continued from previous page

WORD	Base setting	SVD2	SVD3	SVD4	SVD5
question	merely, consider, argue, explain, discuss	argue, explain, discuss, understand, consider	wrong, merely, explain, neither, careful	particular, whole, individual, right, upon	state, argue, consider, discuss, explain
lesson	learn, remember, teach, talk, suvival	remember, remind, explain, teach, learn	curious, remember, learn, teach, creative	think, teach, learn, secret, alone	remember, remind, learn, teach, appreciate
obstacle	avoid, prove, compromise, tackle, face	hinder, importantly, shift, avoid, overcome	inevitably, toward, risky, within, futile	impossible, weak, inevitable, mean, enough	hinder, avoid, overcome, navigate, encounter
victory	score, double, unbeaten, second, triumph	end, beat, win, lose, finish	end, fail, win, finish, manage	win, beat, end, miss, finish	defeat, lose, beat, win, unbeaten
offer	agree, promise, pay, bet, afford	agree, promise, sign, opt, post	agree, ask, willing, reluctant, bet	agree, pay, accept, want, unable	agree, sign, accept, renew, lend
trip	travel, arrive, return, head, cruise	travel, return, visit, arrive, stay	travel, busy, alone, stay, overnight	travel, return, visit, arrive, stay	travel, stay, return, visit, overnight

Continued on next page

Table A.1 – continued from previous page

WORD	Base setting	SVD2	SVD3	SVD4	SVD5
visit	travel, spend, arrive, meet, invite	travel, invite, accompany, spend, meet	afterwards, personally, meet, shortly, immediately	travel, accompany, stay, send, invite	invite, meet, stay, greet, arrange
control	automatic, efficient, manual, require, circuit	allow, direct, maintain, provide, support	effectively, allow, manual, direct, powerful	seize, invade, flee, occupy, gain	maintain, allow, effectively, integrate, manual
invitation	request, award, invite, urge, advise	invite, welcome, visit, arrange, accept	invite, accept, attend, welcome, request	invite, accept, arrange, welcome, abroad	invite, announce, sponsor, solicit, discuss
hint	reveal, reference, recall, evoke, perceive	seem, convey, evoke, curious, reveal	curious, subtly, seem, strangely, genuine	obvious, real, note, evoke, convey	seem, lend, reveal, subtly, reference
demand	consider, offer, expect, need, sufficient	limited, increase, domestic, deal, accept	limited, financial, immediate, mutual, legitimate	expensive, little, poor, less, supply	increase, decline, profitable, expect, urge
challenge	focus, address, concern, deal, understand	gain, aside, fail, claim, address	future, fail, despite, ultimately, progress	good, fail, right, particular, potential, latter	face, decide, lose, claim, address

Table A.2: Similar words computed from various VSMs for French

WORD	Base	SVD2	SVD3	SVD4	SVD5
excuses	pàs, déclarer, pdd, répondre, préciser	navoir, répondre, plaindre, choqué, hésiter	répondre, lintéressé, plaindre, sexcuser, déclarer	navoir, choqué, plaindre, hésiter, prévenir	déclarer, plaindre, répondre, sexcuser, lintéressé
habitude	nhésitant, contentant, quà, semblant, devoir	lhabitude, faire, semblant, quà, répéter	lhabitude, contenter, nhésite, semblant, contentant	lhabitude, faire, quà, semblant, devoir	lhabitude, contente, semblant, lhabitude, nhésite
cours	kamenouchka, chenouf, baalzebul, chaudury, khurasani	jette, prendre, couler, affluent, jusquà	zavant-turent, jusquà, seconde, mondiale, suivre	jette, couler, affluer, prendre, traverser	ensuite, jusquà, lors, zavant-turent, suivre
rêve	lesprit, lenfant, soi, rêver, monstre	lesprit, rêver, imaginer, lhomme, croire	rêver, lesprit, imaginer, croire, étrange	lesprit, rêver, lhomme, raconter, lenfant	rêver, lesprit, imaginer, hanter, lhomme
crime	coupable, commetre, accusé, laffaire, coupables	coupable, commetre, laffaire, punir, accusée	coupable, commetre, accusé, coupables, accusée	coupable, commetre, punir, accuser, laffaire	coupable, commetre, accusé, accuser, coupables

Continued on next page

Table A.2 – continued from previous page

WORD	Base setting	Depth 2 weight	3-2-2	4-3-2	7-1-2
réunion	lassemblée, assyaukanie, diète, congrès, ...	réunir, tenir, dorganiser, congrès, lassemblée	réuni, réunir, lassemblée, tenir, réunis	réunir, dorganiser, congrès, lassemblée, organiser	réunir, réuni, lassemblée, convoquée, dorganiser
colère	voyant, craindre, éprouver, sentir, croyant	voyant, craindre, croyant, sentir, terrible	voyant, éprouver, furieux, sentir, craindre	voyant, croyant, craindre, sentir, sen	voyant, coma, fatigue, léthargie, détresse
sommeil	lenfant, dépuisement, causer, cause, souffrir	mortelle, survenir, souffrir, guérir, mortel	cryogénétique, reparateur, lenfant, réveille, souffrir	mortelle, guérir, mortel, souffrir, lenfant	cryogénétique, lenfant, reparateur, mental, réveille
chanson	mér, siipinä, woulda, unelma, heriotzean	sortir, lalbum, necked, enregistrer, negbeni	labed, tendem, impresionante, härkien, sortir	lalbum, sortir, learn, negbeni, flaha	labed, tendem, sortir, härkien, mistreat

References

Chiu, W. and Lu, K. (2015). Paradigmatic relations and syntagmatic relations: How are they related? *Proceedings of the Association for Information Science and Technology* [online]. 52.1, pp. 1–4. DOI: https://doi.org/10.1002/pra2.2015.1450520100122.

Clark, S. (2015). "Vector Space Models of Lexical Meaning". In: *Handbook of Contemporary Semantics, 2nd edition.*

Fano, R. and Hawkins, D. (1961). Transmission of Information: A Statistical Theory of Communications. *American Journal of Physics*. 29, pp. 793–794.

Hagiwara, M., Ogawa, Y., and Toyama, K. (2008). Effective Use of Indirect Dependency for Distributional Similarity. *Journal of Information Processing*. 15, pp. 19–42.

Heylen, K., Peirsman, Y., Geeraerts, D., and Speelman, D. (2008). "Modelling Word Similarity: an Evaluation of Automatic Synonymy Extraction Algorithms". In: *LREC*.

Jurafsky, D. and Martin, J. H. (2020). *Speech and Language Processing (3rd ed. draft)*.

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, pp. 211–225.

Levy, O. and Goldberg, Y. (June 2014). "Dependency-Based Word Embeddings". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pp. 302–308. DOI: 10.3115/v1/P14-2050. Available from: https://www.aclweb.org/anthology/P14-2050.

Lin, D. (1998). "An Information-Theoretic Definition of Similarity". In: *ICML*.

Lison, P. and Kutuzov, A. (2017). Redefining Context Windows for Word Embedding Models: An Experimental Study. *ArXiv*. abs/1704.05781.

Mel'cuk, I. (1996a). "Lexical functions: a tool for the description of lexical relations in a lexicon". In:

Mel'cuk, I. (Jan. 1996b). Lexical functions in lexicography and natural language processing. *Lexical Functions: A Tool for the Description of Lexical Relations in the Lexicon*, pp. 37–102.

Mel'čuk, I. (2016). *Language: From Meaning to Text*. Academic Studies Press.

Niwa, Y. and Nitta, Y. (1994). "Co-Occurrence Vectors From Corpora Vs. Distance Vectors From Dictionaries". In: *COLING*.

Padó, S. and Lapata, M. (2007). Dependency-Based Construction of Semantic Space Models. *Computational Linguistics*. 33, pp. 161–199.

Padró, M., Idiart, M., Villavicencio, A., and Ramisch, C. (2014). "Comparing Similarity Measures for Distributional Thesauri". In: *LREC*.

Peirsman, Y., Heylen, K., and Speelman, D. (2007). "Finding semantically related words in Dutch: co-occurrences versus syntactic contexts". In:

Pierrejean, B. and Tanguy, L. (June 2018). "Towards Qualitative Word Embeddings Evaluation: Measuring Neighbors Variation". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*. New Orleans, Louisiana, USA: Association for Computational Linguistics, pp. 32–39. DOI: 10.18653/v1/N18-4005. Available from: https://www.aclweb.org/anthology/N18-4005.

Polguère, A. (2016). *Lexicologie et sémantique lexicale: Notions fondamentales*. Presses de l'Université de Montréal. ISBN: 9782760636576. Available from: http://www.jstor.org/stable/j.ctv69t90p.

Pulman, S. (2013). Distributional Semantic Models. In: *Quantum Physics and Linguistics: A Compositional, Diagrammatic Discourse*. Ed. by C. Heunen, M. Sadrzadeh, and E. Grefenstette. Oxford University Press, ISBN 978-0-19-964629-6, pp. 333–358.

Terra, E. and Clarke, C. (Mar. 2004). Frequency Estimates for Statistical Word Similarity Measures. DOI: 10.3115/1073445.1073477.

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