A comprehensive analysis of the parameters in the creation and comparison of feature vectors in distributional semantic models for multiple languages

András Dobó

Supervisor
János Csirik, DSc

Doctoral School of Computer Science
Faculty of Science and Informatics
University of Szeged

A thesis submitted for the degree of
Doctor of Philosophy

Szeged
2019

To my wife Marianna and my daughters Dalma and Hanga

Contents

List of Figures

List of Tables

Abbreviations

NLP	natural language processing
DSM	distributional semantic model
CVBM	count-vector-based model
PM	predictive model
MF	full MEN dataset
MD1, MD2	the two development parts of the MEN dataset
MT	the test part of the MEN dataset
RG	RubensteinGoodenough-65 dataset
MC	MillerCharles-28 dataset
WS	WordSim-353 dataset
SL	SimLex-999 dataset
TO	TOEFL dataset

Ew	the 26.05.2011 dump of the English Wikipedia
Dc	the information extraction method of Dobó and Csirik (2013)
Lc	the information extraction method of Levy et al. (2015)
Ec	the information extraction method of Salle et al. (2016a)
Mv	the semantic vectors of Mikolov et al. (2013b)
Bv	the semantic vectors of Baroni et al. (2014)
Pv	the semantic vectors of Pennington et al. (2014)
Sv	the semantic vectors of Speer et al. (2017)
Ev	the semantic vectors of Salle et al. (2018)
P	Pearson's correlation
S	Spearman's correlation
H	modified harmonic mean of P and S
A	accuracy
BSS	basic settings set
CPS	a specific combination of parameter settings for a model
configuration	a specific combination of parameter settings for a model
model	a system with specific configuration using specific input data
XUsingY	a specific model with configuration X using Y as input data

Acknowledgements

First of all, I would like to express my deepest gratitude to my wife Marianna for all her support. She has always been a great motivation to me and has been by my side at all times. Furthermore, I would like to thank both her and my daughters Dalma and Hanga for all the love and joy I received from them, and all the patience they showed towards me throughout pursuing my PhD.

Moreover, I would like to thank my PhD supervisor, János Csirik, for supervising my work. His advice, as well as his constructive and often critical comments have always been of great value to me.

Additionally, I am grateful to my master's supervisor, Stephen Pulman, for raising my interest in Natural Language Processing through his exciting Computational Linguistics lectures and for helping me kick-start my research in this field while doing my MSc in Oxford.

Last but not least, I would like to say thank you to my parents for supporting me throughout my education.

Abstract

Measuring the semantic similarity and relatedness of words is important for many natural language processing tasks. Although distributional semantic models designed for this task have many different parameters, such as vector similarity measures, weighting schemes and dimensionality reduction techniques, there is no truly comprehensive study simultaneously evaluating these parameters while also analysing the differences in the findings for multiple languages.

We would like to address this gap with our systematic study by searching for the best configuration in the creation and comparison of feature vectors in distributional semantic models for English, Spanish and Hungarian separately, and then comparing our findings across these languages.

During our extensive analysis we test a large number of possible settings for all parameters, with more than a thousand novel variants in case of some of them. As a result of this we were able to find such configurations that significantly outperform conventional configurations and achieve state-of-the-art results.

CHAPTER 1

Introduction

For many natural language processing (NLP) problems, including information retrieval (Hliaoutakis et al., 2006), spelling correction (Budanitsky and Hirst, 2001) and noun compound interpretation (Dobó and Pulman, 2011) among many others, it is crucial to determine the semantic similarity or semantic relatedness of words. While relatedness takes a wide range of relations between words (including similarity) into account, similarity only considers how much the concepts denoted by the words are truly alike. Thus similarity entices relatedness, but not vice versa. For example, the words "bicycle" and "motorbike" are similar, as both denote 2-wheeled vehicles, and thus they are also related. On the other hand, the words "postman" and "mail" are highly related, as usually mails are delivered by postmen, and yet they are not similar, as they denote rather different concepts.

Further, the words "furnace" and "voyage" are neither similar nor related. For a detailed discussion about meaning, relatedness and similarity please refer to Section 2.

1.1 Motivation

Most models are based on the distributional hypothesis of meaning (Harris, 1954), and thus calculate this similarity or relatedness using distributional data extracted from large corpora. These models can be collectively called as distributional semantic models (DSMs) (Baroni and Lenci, 2010; Baroni et al., 2014). In these models first possible features are identified, usually in the form of context words, and then a weight is assigned for each word-feature pair using complex methods, thus creating feature vectors for all words. The similarity or relatedness of words are then calculated by comparing their feature vectors using vector similarity measures. Although DSMs have many possible parameters, a truly comprehensive study of these parameters, also fully considering the dependencies between them, is still missing and would be needed, as also suggested by Levy et al. (2015).

Most papers presenting DSMs focus on only one or two aspects of the problem, and take all the other parameters as granted with some standard setting. For example, the majority of studies simply use cosine as vector similarity measure (e.g. Bruni et al., 2013; Baroni et al., 2014; Speer et al., 2017; Salle et al., 2018) and/or (positive) pointwise mutual information as weighting scheme (e.g. Islam and Inkpen, 2008; Hill et al., 2014b; Salle et al., 2018) out of convention. And even in case of the considered parameters, usually only a handful of possible settings

are tested for. Further, there are also such parameters that are completely ignored by most studies and have not been truly studied in the past, not even separately (e.g. smoothing, vector normalization or minimum feature frequency). What's more, as these parameters can influence each other greatly, evaluating them separately, one-by-one, would not even be sufficient, as that would not account for the interaction between them.

There are a couple of studies that consider several parameters with multiple possible settings, such as Lapesa and Evert (2014) and Kiela and Clark (2014), but even these are far from truly comprehensive, and do not fully test for the interaction between the different parameters. So, although an extensive analysis of the possible parameters and their combinations would be crucial, as also suggested by (Levy et al., 2015), there has been no research to date that would have evaluated these truly comprehensively. Moreover, despite the fact that the best parameter settings for the parameters can differ for different languages, the vast majority of papers consider DSMs for only one language (mostly English), or consider multiple languages but without a real comparison of findings across languages. In this thesis we would like to address these gaps.

1.2 Aims and objectives

DSMs have two distinct phases in general. In the first phase statistical information (e.g. raw counts) is extracted from raw data (e.g. a large corpus of raw text), in the form of statistical distributional data. In the second phase, feature vectors are created from the extracted information for each word and these vectors are then compared to each other to calculate the similarity or relatedness of words.

In our study we take the distributional information extracted in the first phase as already granted, and present a systematic study simultaneously testing all important aspects of the creation and comparison of feature vectors in DSMs, also caring for the interaction between the different parameters.

We have chosen to only study the second phase of the DSMs, as the two phases are relatively distinct and independent from each other, and testing for every single possible combination of the parameter settings in the second phase is already unfeasible due to the vast number of combinations. So instead of a full analysis we already had to use a heuristic approach. Thus also trying to test for the parameters of the first phase (e.g. source corpus, context type (window-based or dependency-based) and context size) simultaneously would be unreasonable and unmanageable, and is out of scope of this study. Therefore we have omitted the examination of this phase completely, with one exception to this.

DSMs relying on information extracted from static corpora have two major categories, based on the type of their first phase: count-vector-based (CVBM) and predictive models (PM; also called word embeddings) (Baroni et al., 2014). In order to get a more complete view and due to the huge popularity of predictive models in recent years, in addition to using information extracted from a corpus using a count-vector-based model, we have also done some experiments with information extracted by a predictive model in case of English. Further, later on we also extended our analysis with a model based on semantic vectors constructed from a knowledge graph. Our intuition was that there will be a single configuration that achieves the best results in case of all types of models. However, please note that in the latter case only a part of the considered parameters could be tested for due to the characteristics of such models. That is part of the reason

why we have focused on count-vector-based DSMs more.

During our research we have identified altogether 10 important parameters for the second phase of count-vector-based DSMs, such as vector similarity measures, weighting schemes, feature transformation functions, smoothing and dimensionality reduction techniques. However, only 4 of these parameters are available when predictive or knowledge-graph-based semantic vectors are used as input, as in case of such input the raw counts are not available any more, the weighted vectors are already constructed and their dimensions are usually also reduced.

In the course of our analysis we have simultaneously evaluated each parameter with numerous settings in order to try to find the best possible configuration (configuration) achieving the highest performance on standard test datasets. We have done our extensive analysis for English, Spanish and Hungarian separately, and then we have compared our findings for the different languages.

For some of the tested parameters a large number of possible settings were tested, more than a thousand in some cases, resulting in trillions of possible combinations altogether. While of course also testing the conventionally used parameter settings, we also proposed numerous new variants in case of some parameters. Further, we have tested a vast number of novel configurations, with some of these new configurations considerably outperforming the standard configurations that are conventionally used, and thus achieving state-of-the-art results.

First we have done our analysis for English and evaluated the results extensively (Dobó and Csirik, 2019a). Then we have repeated the same analysis, with an increased number of settings for several parameters, for English, Spanish and Hungarian, and compared the findings across the different languages (Dobó and

Csirik, 2019b).

For reproducibility and transparency, we plan to make our most important data, code and results publicly available at:

https://github.com/doboandras/dsm-parameter-analysis/.

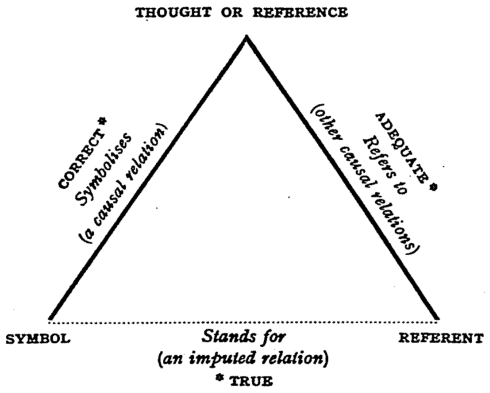
CHAPTER 2

Background

2.1 Meaning

Even in ancient times some great thinkers started to deal with semantics, that is the study of meaning, and philosophized about the definition of meaning. Most notably, Plato's Theaetetus (Burnyeat et al., 1990) is devoted to the nature of knowledge. His view, still central to semantics, was that one knows something if they can account for its details (Kornai, 2019). Later on, Aristotle declared that the meaning of things is a result of convention (Chernyak, 2017). Further, the semiotic triangle of symbol, referent and thought/reference (triangle of meaning) (see Figure 2.1), published in Ogden and Richards (1923), can be traced back as far as Aristotle's De Interpretatione (Ackrill, 1975).

Figure 2.1. The semiotic triangle of symbol, referent and thought/reference (triangle of meaning), taken from Ogden and Richards (1923).



The study of the field of semantics also continued through modern times, with many principles and theories of meaning articulated by noted philosophers and linguists alike. One of the most important of these, the principle of contextuality, formulated by Frege (1884), states that the meaning of words cannot be studied in isolation, rather only in the context of a sentence. Another important concept, the principle of compositionality (also called Frege's principle), articulating that the meaning of a complex expression is defined by its syntactic structure and the meaning of its parts, is also widely credited to Frege (Kornai, 2019). However, recently Pelletier (2001) and Janssen (2001) argued that this was not explicitly stated by Frege himself, and was actually a misinterpretation of Frege's thoughts to some extent. Further, this idea actually also appeared in many previous works, even as early as Plato's Theaetetus (Burnyeat et al., 1990).

Later on, Saussure (1916) considered language as a system of signs expressing ideas, where a sign is a composition of a signifier (significant) and a signified (signifié), and thought that the relation between these two components is of arbitrary nature. Wundt (1920) believed that mental contents receive their mean-

ing through their relation to other mental contents , which is usually referred to as the principle of relational analysis or the context principle. Then, Katz and Fodor (1963) was convinced that word meaning is made up of a collection of semantic markers and a distinguisher. In this theory the formal part of meaning is defined by the semantic markers, determining the semantic properties of expressions, while the unsystematic distinguisher disposes of the semantic residue.

There are many different, often contradictory perspectives in semantics. Langacker (2008) gives a very good overview of some of the possible different views of meaning. According to the cognitive linguistic position, the meaning of an expressions is in the mind of the speaker producing and understanding it. In sharp contrast to this, there are some views that completely ignore the human mind and body: the platonic view sees language as an abstract, unlocalizable entity, while the objectivist perspective defines the meaning of an expressions with those conditions under which it is true.

On the other hand, Langacker (2008) himself sees the interactive view as more reasonable. This perspective again takes humans into account, but views meaning as dynamically changing through discourse and social interaction instead of being fixed and predetermined in one's mind. In this respect meaning is not viewed as localized to one human mind any more, but rather as being distributed in the speech community, in the context and in the surrounding world.

To take another different view, formal semantics treats natural languages the same way as formal languages, and tries to define meaning by constructing precise mathematical models between expressions and real-world entities (Aronoff and Rees-Miller, 2003).

To look at a more practice-oriented perspective, distributional semantics is

based on the distributional hypothesis of meaning, which states that words occurring in similar contexts tend to have similar meaning (Harris, 1954). Following on this idea, Firth (1957) argued that one can get to know the meaning of a word by recognizing in what contexts it occurs. Most currently used models of meaning are actually based on this hypothesis in practice, and represent words with vectors based on distributional (contextual) data.

There are also such perspectives that mix practice-oriented views with theoretical ones, trying to combine the advantages of both. One such recent approach worth mentioning is that of the sparse overcomplete word vector representations proposed by Faruqui et al. (2015). This combines the interpretable features from the theory of lexical semantics with the (usually dense) word vectors from distributional models to come up with sparse (and optionally binary) word vectors, resembling the interpretable features from lexical semantics.

2.2 Semantic similarity and relatedness

After defining a representation of meaning, it is possible to study the relations between the meanings of words. There are many types of relations that can exist between the meanings of two words, including hyponymy, hypernymy, synonymy and antonymy, among many others (Brinton, 2000). Based on these relations, it is possible to evaluate the strength of the semantic relationship (association) between words. Semantic relatedness takes any relation between the words into account (including semantic similarity), thus assessing how close the concepts denoted by the words are to each other with respect to any type of relation. On the other hand, semantic similarity is more specific, and is only concerned about

how much the concepts denoted by the words are truly alike, thus only taking the subsumption ("is a") relation into account (Harispe et al., 2015; Banjade et al., 2015). For better understanding of these two notions, a couple of examples were presented in the first paragraph of Section 1.

Based on the different views and definitions of meaning, one can define similarity multiple ways. If the meaning of words is believed to be located in the mind, then one could represent concepts with points in a mental space. Thus, in the mental distance approach similarity between words can be defined as some kind of distance in this mental space (Shepard, 1962). On the other hand, in those views, where concepts are represented with the help of lists of features, one could compare the meaning of words by analyzing the commonalities and differences in the list of features of the words' concepts (featural approach) (Tversky, 1977). There are also such approaches that could be applied to any type mental representation. For example, in the transformational approach any mental representation can be transformed into another one, and the similarity of words can be based on the transformational steps needed to transform the concept of one word to another (Hahn et al., 2003).

On the other hand, as stated before, most current models of meaning used in practice are based on the distributional hypothesis (Harris, 1954). In these distributional semantic models (DSMs), words are considered similar if they occur in similar contexts, based on the definition of the hypothesis itself.

2.3 Distributional semantic models

As determining the semantic similarity and relatedness of words can be important for many NLP problems, research into methods automatically determining this have started decades ago. Therefore there already exist a vast number of systems for this task, and one can distinguish many different types among them. The most logical and usual way to categorize these systems is based on the type and usage method of input data employed in these systems.

Most current DSMs rely solely on linguistic data, as one would expect. They usually take large corpora as input, from which they extract statistical information, but there are also numerous systems making use of other linguistic input.

Static corpora are widely used as input due to their easy usage and wide availability. As for DSMs usually simple raw text is used as input, so manual annotation of the text is not needed. This makes it is easy to generate or acquire such input data for a wide range of languages and topics. Hence, DSMs using such input can usually be easily adapted to different languages and domains. As we have already discussed before, there are two main categories of methods based on information extracted from static corpora: count-vector-based and predictive models.

A typical count-vector-based models is that of Pennington et al. (2014), who create word vectors by combining global matrix factorization with local context window method, and then training only on nonzero elements in the co-occurrence matrix. Levy et al. (2015) improve previous count-vector-based models with ideas taken from predictive models, thus improving their performance significantly. With this they show that contrary to previous belief, count-vector-

based models can perform as well as predictive models with the right system design choices and hyper-parameter optimizations.

Iosif et al. (2016) improve previous state-of-the-art results by presenting a novel, cognitively motivated type of DSMs, motivated by the dual-processing cognitive perspective in short-term human memory. Salle et al. (2016a), Salle et al. (2016b) and Salle et al. (2018) propose a model of distributed word representations by performing explicit, stochastic factorization of the positive pointwise mutual information matrix, and then present several enhancements to their original system. We have found their final results to be the current state-of-the-art for calculating semantic relatedness among the count-vector-based models.

One of the most well known examples of predictive models are those of Mikolov et al. (2013a) and Mikolov et al. (2013b). Both the continuous bag-of-words (CBOW) and continuous skip-gram (Skip-gram) learning algorithms for these models are publicly available in the word2vec toolkit[1], which has become very popular and has been used in numerous systems since its publication. For example, Baroni et al. (2014) perform an extensive comparison of CBOW predictive models with traditional count-vector-based models based on Collobert and Weston (2008) and Baroni and Lenci (2010), and found their predictive models to perform consistently better. Similarly, De Deyne et al. (2017) also use the CBOW approach with settings based on previous work of others to achieve close to state-of-the-art results.

There are also many models that combine multiple input data into a single model, using different techniques for this combination, ranging from sim-

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

ple methods to complex machine learning techniques. Yin and Schütze (2016) propose to learn metaembeddings by combining multiple embedding sets in an ensemble approach, thus increasing vocabulary coverage and achieving better performance than by using the embedding sets individually. Similarly to this, Christopoulou et al. (2018) also propose a mixture of multiple models. However, they propose to create multiple topic-specific DSMs based on topic-specific subcorpora, and then combine the scores of the different models for a word pair into a single score, to achieve state-of-the-art results.

There are also many models, that beside or instead of corpora, take other types of linguistic resources as input. Many of these make use of large lexical databases (e.g the WordNet (Fellbaum, 1998)), knowledge graphs (e.g. the ConceptNet (Speer et al., 2017)), concept lexicons (e.g. the 4lang concept lexicon (Kornai, 2010)), word association datasets (e.g. the Small World of Words project word association dataset (De Deyne et al., 2017)), or other similar datasets. These datasets are usually mostly or completely hand-crafted, often by experts, therefore they have very high quality. On the other hand they are rather expensive to create, have limited coverage and they are costly to update as languages evolve. Further, methods based on such datasets cannot easily be adapted to other languages or domains without having similar datasets for the other languages and domains too.

Recski et al. (2016), for example, beside word embeddings, also make use of the WordNet and the 4lang concept lexicon, to achieve state-of-the-art word similarity results. Lee et al. (2016) and Rothe and Schütze (2017) also combine word embeddings with the WordNet to achieve rather good results on word relatedness tasks. De Deyne et al. (2017) use a spreading activation approach and per-

form a random walk on their Small World of Words project word association dataset to achieve very good word relatedness results. Finally, (Speer et al., 2017) combine word embeddings with their ConceptNet knowledge graph to achieve overall state-of-the-art word relatedness performance.

Other models also take advantage of the vast amount of text present on the World Wide Web, and exploit these data by issuing web search queries on commercial search engines. These models have the advantage of having access to huge amount of text, however, have many disadvantages due to numerous limitations and drawbacks posed by commercial search engines (Nakov, 2007; Kilgarriff, 2007). For example, Kulkarni and Caragea (2009) create concept clouds for words, and then compares these concept clouds to determine the relatedness of words. For both parts they issue web search queries. Yih and Arbor (2012) propose a combination of heterogeneous vector space models, also including ones based on web search results. On the other hand, Iosif and Potamianos (2015) propose the use of web search engines in a rather unconventional way: they issue targeted web queries to create a corpus, which then can be used as input for their model.

Further, there are also such models that employ other types of input beside linguistic data, such as images. For example, Bruni et al. (2013) present a novel approach by using images to create "visual words", and using these alongside linguistic input in their model. Lazaridou et al. (2015) also make use of visual information and combine it with a Skip-gram model. Collell et al. (2017) present a language-to-vision mapping, and uses the output visual predictions of this mapping in their multimodal embeddings model to achieve rather good word relatedness performance.

Although even some early DSMs have experimented with trying multiple settings for one or more parameters, such as vector similarity measures (Jones and Furnas, 1987), even in most of the current state-of-the-art systems, both those only using distributional linguistic data and those making use of other types of data too, as well as either count-vector-based, predictive or knowledge-graph-based models, actually only one similarity measure (predominantly cosine similarity) was tested (e.g. Yih and Arbor, 2012; Bruni et al., 2013; Baroni et al., 2014; Hill et al., 2014a; Pennington et al., 2014; Wieting et al., 2016; Faruqui and Dyer, 2015; Lazaridou et al., 2015; Banjade et al., 2015; Levy et al., 2015; Iosif et al., 2016; Salle et al., 2016a; Rothe and Schütze, 2017; Collell et al., 2017; De Deyne et al., 2017; Speer et al., 2017; Salle et al., 2018; Christopoulou et al., 2018; Vakulenko, 2018). Further, most count-vector-based DSMs only test for one weighting scheme (e.g. Islam and Inkpen, 2008; Yih and Arbor, 2012; Bruni et al., 2013; Hill et al., 2014b; Levy et al., 2015; Iosif et al., 2016; Salle et al., 2016a, 2018), mainly based on pointwise mutual information (PMI) (Church and Hanks, 1990) in almost all cases. Moreover, many of the other possible parameters, such as feature transformation, smoothing, dimensionality reduction or filtering stop words, have been completely neglected in the vast majority of studies.

Of course there are also a couple of studies that try to examine one or more of the parameters of DSMs in detail. Some of them focus solely on vector similarity measures, neglecting all other aspects of the systems, with Jones and Furnas (1987) and Weeds (2003) testing several different settings. On the other hand, instead of vector similarity measures, Evert (2005) and Pecina (2010) evaluate different weighting schemes extensively. There are also studies with respect to vector comparison methods, outside the domain of NLP (either general studies

or from some other domain) that deal with either vector similarity measures (e.g. Cha, 2007; Deza and Deza, 2016) or vector weighting schemes (e.g. Zhang et al., 2011) extensively, without considering any other aspects of vector comparison.

A handful of studies, such as Curran (2004), Lapesa and Evert (2014) and Kiela and Clark (2014), beside considering multiple vector similarity and weighting scheme settings, also take other parameters into account, like feature transformation or dimensionality reduction, considering a few of the possible settings for them. These more complex studies sometimes also mix the parameters of the information extraction (first phase of DSMs) with those of the creation and comparison of feature vectors (second phase of DSMs) and thus also include parameters like source corpus, context type and context size. But even these complex studies usually neglect many other important aspects of the problem, do not account for the interaction between the different parameters sufficiently, and/or only test for a handful of different settings for each parameter. So evaluating all the possible parameters together and testing their possible combinations extensively would be crucial, but has not been addressed sufficiently yet.

Moreover, most models were only tested for English and neglect any other languages despite the fact that DSMs might work differently across multiple languages. Of course, there are several studies in which results were presented for languages other than English, including Spanish (Hassan and Mihalcea, 2009; Moldovan et al., 2015; Camacho-Collados et al., 2017) and Hungarian (Dobó and Csirik, 2012; Novák and Novák, 2018). However, even those that include multiple languages usually only present some test results for the different languages separately, without any real analysis of the differences in the findings between the languages. Furthermore, for Hungarian there did not previously exist any stan-

dard evaluation datasets. For example, the models of Novák and Novák (2018) were evaluated manually by experts. Therefore, to be able to have a reproducible and standardized evaluation method, we have created Hungarian test datasets in Dobó and Csirik (2012) and in Dobó and Csirik (2019b), and evaluated our models on these. For reproducibility and transparency, we have included the Hungarian test datasets in Appendix C.

CHAPTER 3

Data and evaluation methods

3.1 Data

We focused on the second phase of DSMs, so our analysis took information extracted from a corpus as granted. As already stated above, we wanted to focus our attention mostly on count-vector-based models, but also wanted to experiment with predictive and knowledge-graph-based models a little. A vast number of configurations needed to be tested, as detailed in the next chapter, therefore we had to choose a relatively small corpus for information extraction in case of the count-vector-based models. Finally, for English we have chosen the British National Corpus (BNC; (BNC Consortium, 2001)), a rather small (about 100 million words) but balanced corpora, from which the information was extracted by the

bag-of-words method presented in Dobó and Csirik (2012) and Dobó and Csirik (2013).

This information extraction method finds each occurrence of the selected word in the used corpora, then includes every word in a window of 3 words within that occurrence in the feature vector. However, it is different from regular bag-of-words approaches, since it counts the occurrences of close words multiple times. Specifically, the frequency this method assigns to a feature word is based on the distance of the observed word and the feature word. Several different techniques were tested, and the best was found to be using frequencies that scale quadratically with the distance (with a window size of 3, frequency 9 is assigned to distance 1, frequency 4 to distance 2 and frequency 1 to distance 3).

Here the extracted raw counts were used, which will be referenced as Dobó and Csirik's counts on the BNC (DcBnc) in the rest of the thesis. As this extraction method extracts information for nouns, verbs, adjectives and adverbs separately, our model had to guess the part-of-speech of the words in the used datasets before comparing them. We have used the original method presented in Dobó and Csirik (2012) and Dobó and Csirik (2013) to guess the part-of-speech (POS) of input words when creating their feature vectors. The POS of words in a question can be inferred from the other words contained in the same question. For our methods, we assumed that each input word is a verb, noun, adjective or adverb and each question contains words of the same POS. For a question the part-of-speech maximizing the following formula was chosen:

$$pos = argmax_p \prod_{w \in q} ln(1.0001 + f_{w,p}) \tag{3.1}$$

where p can take any of the four possible POSs, q denotes the question, w runs through the words of q and $f_{w,p}$ is the frequency of w having part-of-speech p.

For Spanish and Hungarian, we have chosen the similarly sized Spanish Wikicorpus (Reese et al., 2010) (EsWiki, about 110 million words) and the 23.01.2012 dump of the Hungarian Wikipedia (HuWiki; about 65 million words), respectively, and employed the same information extraction method.

For the predictive model, the size of the corpus was much less relevant with respect to the feasibility of our analysis, as much fewer parameters were tested, and the number of dimensions of the feature vectors was also small. We have decided on the most widely used dataset, the Google News corpus (GNC) of around 100 billion words, from which feature vectors for words were created by Mikolov et al. (2013b). These 300-feature-long word vectors contain real weights for all features, and the vectors are already L_2 normalized.[1] These semantic vectors will be referenced as Mikolov et al.'s vectors (Mv) from now on.

In case of the knowledge-graph-based models, we have decided to experiment with the state-of-the-art model of Speer et al. (2017) (Sv), which is based on the ConceptNet.

For some final tests we have also used the text of the 26.05.2011 dump of the English Wikipedia (Dobó and Csirik, 2013) (Ew; about 1.2 billion words), the ukWaC corpus (Baroni et al., 2009) (about 2 billion words), raw counts obtained using the information extraction method of Levy et al. (2015)[2] (Lc) and of Salle et al. (2016a)[3] (Ec), as well as the semantic vectors of Baroni et al. (2014) (Bv),

[1] The word vectors are publicly available at https://code.google.com/archive/p/word2vec/.
[2] https://bitbucket.org/omerlevy/hyperwords/
[3] https://github.com/alexandres/lexvec/

Pennington et al. (2014) (Pv) and Salle et al. (2018) (Ev).

3.2 Evaluation

To be able to compare the performance of the different configurations in English, we have primarily chosen to employ the MEN dataset (Bruni et al., 2013), whose test part (MT) was used as a test set and whose development part was split into two equal chunks randomly to get two development sets (MD1 and MD2). Moreover, to present a comprehensive evaluation, we have decided to also test our measures on all the other commonly used datasets, namely the RubensteinGoodenough-65 (RG; (Rubenstein and Goodenough, 1965)), the Miller-Charles-28 (MC; (Resnik, 1995)), the WordSim-353 (WS; (Finkelstein et al., 2002)), the SimLex-999 (SL; (Hill et al., 2015)) and the TOEFL (TO; (Landauer and Dumais, 1997)) datasets too. However, the MC, RG, WS and TO datasets were mostly included because of their wide use in previous decades. As their size is relatively small and the results on them are rather unreliable, as also noted by Camacho-Collados et al. (2017), conclusions based on these have to be taken cautiously. As some researchers have used the full MEN dataset (MF) for testing, we have also evaluated our best methods on this for comparability with the results of others. However, please note that our results are not fully reliable on this dataset, as two-thirds of it has already been used in the process of determining the best possible configurations.

In case of Spanish and Hungarian, we have made use of the Spanish WordSimilarity-353 (WSEs; (Hassan and Mihalcea, 2009)), the Moldovan (MOEs; (Moldovan et al., 2015)) and the Spanish Rubenstein Goodenough (RGEs; (Camacho-Collados

et al., 2015)) datasets for Spanish, and parts of the Hungarian version of the TOEFL (TOHu1 and TOHu2; (Dobó and Csirik, 2012)) and Rubenstein Goodenough data-sets for Hungarian (RGHu; Dobó and Csirik (2019b)). The last was constructed the same way as the Hungarian TOEFL and Miller Charles datasets in Dobó and Csirik (2012). For reproducibility and transparency, we have included the Hungarian test datasets in Appendix C.

However, we have to note that all of the used Spanish and Hungarian datasets are rather small and except for the Moldovan dataset just translated from English datasets, which can distort them. The Hungarian datasets are especially small, and the type of the TOEFL dataset also makes the results on it even less reliable compared to the other datasets. However, due to the lack of truly suitable resources, we had to settle for these.

The TO, TOHu1 and TOHu2 datasets include questions, the task being the selection of the most similar word from the four answers to the question word. Here the accuracy (A) of the models in case of the similarity questions can be used for evaluation purposes.

All other datasets include word pairs with gold standard scores (the last one for similarity, the other ones for relatedness) assigned to them by human annotators. For such datasets two standard evaluation techniques are widely used, namely calculating the Pearson product-moment correlation coefficient (P) and the Spearman's rank correlation coefficient (S) between the gold standard scores and the scores returned by the evaluated model. Some previous studies report both, with others only one or the other, with a significant preference for the S. Because we think that both of them are important and meaningful, especially as during our tests we have experienced that many models achieving either high

P or high S score performed terribly with respect to the other score, we have decided to use both during our analysis. Further, to be able to take both of them into account in a single measure, we have created a modified harmonic mean measure of the two coefficients, as follows:

$$H(P,S) = \frac{2 \times P \times S}{|P| + |S|} \tag{3.2}$$

The original version of the harmonic mean of P and S has been previously used by Aouicha et al. (2016) too for the same reason. However, without our modification it can result in very high scores if the magnitude of either P or S is just a little larger than the other and they have different signs, which property is very undesirable. As opposed to this, our version can also handle negative arguments properly, returns a negative score in all cases where P and S have different signs, and keeps the codomain [-1,1] of P and S.

Further, during the first part of our analysis the best performing parameter settings had to be selected based on multiple runs for each setting. In this process we have employed the following measures:

- MaxP, MaxS and MaxH, for the maximum of P, S and H measures achieved during the multiple runs of the given parameter setting, respectively
- AvgP, AvgS and AvgH, for the average of P, S and H measures achieved during the multiple runs of the given parameter setting, respectively
- T10P, T10S and T10H, for the proportion of the runs of the given parameter setting with performance in the top 10%, out of all runs of all considered settings of that parameter, based on the P, S and H measures, respectively

However, we have only reported the MaxS and MaxH scores in the rest of the thesis for easier readability.

Due to the large number of abbreviations used in this thesis, we have decided to summarize them at the beginning of this thesis to make the reading easier.

CHAPTER 4

The general description of our analysis

4.1 The heuristic approach

The task was to try to find the best possible configuration, considering every possible setting of all the considered parameters (10 parameters in case of count-vector-based DSMs and 4 in case of predictive and knowledge-graph-based DSMs). However, as the number of possible combinations are in the magnitude of trillions in case of count-vector-based DSMs, it would have been unfeasible to test every single combination one-by-one with our limited resources. Instead of this full analysis, we chose a heuristic approach to search for the best configuration, which consisted of two phases.

Prior to the first step a basic set of a handful of parameter settings (BSS) was

created for each parameter in such a way, so that the selected settings in each set should achieve good performance on some preliminary tests, while they should be as different from each other as possible and their collection should give as good a representation of the set of all settings of the given parameter as possible. Further, the most commonly used settings were also always selected in case of each parameter.

In the first phase, each parameter was tested separately on the first development dataset (MD1), in order to select a candidate list of settings of the given parameter for the second phase. For this selection process, in case of each parameter, all such configurations were tried, where the settings came from all the possible settings in case of the tested parameter, and from the basic settings set in case of the other parameters. This reduced the number of possible combinations exponentially and thus (by restricting the number of settings for the other parameters) made it possible to test all the possible settings of the given parameter. Based on the tests, such (preferably diverse) settings were selected for the second phase that seemed to be the most promising and most likely to be part of the ultimate best configuration. This selection was done based on the 9 measures (Max*, Avg*, T10*) introduced for this task in Section 3.2. To be able to select as many different types of measures, we have tried to avoid selecting too many very similar measures into the second phase, and rather selected diverse measures. Further, some conventionally used settings from the past decades were also included in the second phase irrespective of their performance, to make comparison with conventional configurations in general use easier.

After this, in the second phase, tests with all combinations of the selected settings for all parameters were conducted on the second development dataset

(MD2). In case of count-vector-based DSMs this was done instead of a full analysis, as that would have been unfeasible due to the vast number of combinations. In case of predictive and knowledge-graph-based DSMs only 4 out of the 10 parameters could be tested due to the characteristics of such models, so there were much less possible configurations than in case of count-vector-based DSMs. This made a full analysis of all possible configurations also feasible in Dobó and Csirik (2019a), when the number of tested settings for these 4 parameters were still considerably lower than now. We have decided to also do this then beside the heuristic analysis in the hope of being able to further validate both our idea of the heuristic method for selecting the best configuration and our results (see Section 5.2.3).

That configuration was selected as best, which achieved the best H value on the MD2 dataset. The selected best measure was then evaluated on all test datasets.

The idea behind our heuristic approach was that hopefully the ultimate best configuration is composed of such parameter settings that also seem to be promising in general, and when tested separately, thus achieving good performance in the first phase too. This heuristic approach limits the number of needed runs exponentially compared to the full test of all possible settings for every parameter, while hopefully resulting in the same or at least very similar outcome.

First we have done this two-step heuristic analysis for English and evaluated the results extensively (Dobó and Csirik, 2019a). Then we have repeated the same analysis, with an increased number of settings for several parameters, for English, Spanish and Hungarian, and compared the findings across the different languages (Dobó and Csirik, 2019b).

4.2 The tested parameters

During our research, we have identified 10 parameters as possibly important in the creation and comparison of feature vectors in DSMs. In this section we will give a detailed introduction to these parameters as well as to their possible settings, with their abbreviation and number of possible settings tested for them in parentheses after their name. In case of predictive and knowledge-graph-based models, only 4 out of the 10 parameters could be tested due to the characteristics of such models. These are marked with * after their name. Further, in case of 2 of these 4 parameters, there were some settings that had to be discarded for similar reasons.

Although presenting the definition of all the settings for the semantic similarity and weighing scheme parameters would be impossible here due to their large number, we have included the most important formulas in Appendices A and B. Further, for reproducibility and transparency, we plan to make the list of all the tested settings for these parameters, together with their respective formula, references and achieved results publicly available at:

https://github.com/doboandras/dsm-parameter-analysis/.

4.2.1 Vector similarity measures* (VecSim; 1221)

There are two general methods for comparing vectors: calculating their similarity or the difference between them. In order to be able to evaluate all measures consistently, all distance measures have been converted to similarity measures as follows:

$$s(u,v) = \frac{1}{1 + |d(u,v)|} \tag{4.1}$$

The same conversion has also been used by Kiela and Clark (2014), and its inverse (similarity to distance conversion) by Deza and Deza (2016), but both of them without the absolute signs.

Vector similarity measures are then used for the comparison of the feature vectors of words, to produce the (final) similarity score of the words. They are an essential part of DSMs, and have been evaluated in many previous studies (e.g. Weeds, 2003; Curran, 2004; Lapesa and Evert, 2014; Kiela and Clark, 2014), as also noted in Chapter 2.

Many of the similarity measures have both a numerical and a binary variant. To make things easier we have decided not to explicitly implement any binary versions. Instead of explicitly implementing these binary variants too, we have implemented a binary weighting scheme, called identity. Using this weighting scheme essentially converts the numerical similarity measures to binary ones. This has greatly reduced the number of similarity measures that had to be explicitly implemented, while implicitly also testing them.

Altogether 1221 variants have been tested, which include:

- simple measures based on the inner product (e.g. cosine similarity (Jones and Furnas, 1987) and harmonic mean (Cha, 2007)),
- measures of correlation (e.g. Pearson correlation (Jones and Furnas, 1987)),
- statistical coefficients (e.g. Dice coefficent (Kiela and Clark, 2014) and Jaccard index (Curran, 2004)),

- measures of Minkowski distance and others related to this (e.g. L_1 distance (Cha, 2007) and Sorensen distance (Deza and Deza, 2016)),
- measures designed for comparing probability distributions (e.g Jensen-Shannon divergence (Cha, 2007)),
- measures of statistical hypothesis testing (e.g. χ^2 test (Cha, 2007)),
- and their many-many variants,
- as well as numerous new measures.

Actually, more than 90% of these were new measures proposed by us, with most of them based on some commonly used measure. Our idea was that the measures we used as basis are rather simple, but they still perform quite well. So we thought that adding some further sophistication to them might improve on the already good results. The measures used as basis include the inner product (InnerProd) (Jones and Furnas, 1987), the cosine similarity (Cos) (Jones and Furnas, 1987), the Pearson correlation (Pears) (Jones and Furnas, 1987), the Minkowski (L_p) distances (Cha, 2007), the Penrose shape distance (PenroseShape) (Deza and Deza, 2016), the Maryland Bridge similarity (Mb) (Deza and Deza, 2016), and Lin's similarity measure (Lin) (Lin, 1998a), among others. These were usually modified using some weighting or transformation function inside or outside the summation in them, or by trying out different versions for their normalization factors.

There are also a large number of new variants combining the features of already existing versions of Cosine similarity and alike measures, such as the Pearson (Pears) (Jones and Furnas, 1987), the Adjusted cosine (AdjCos) (Shalaby and

Zadrozny, 2016), the Maryland Bridge (Mb) (Deza and Deza, 2016) and the PF-Mod (Pascual and Fujita, 2017) measures. For some other new measures the intuition came from other fields, such as from signal processing (Scharf and Demeure, 1991) for the sum of the ratio of signal and noise (SRSN) or from statistics (Wonnacott and Wonnacott, 1990) for the standard deviation-based (STDLike) measure. The number of possible settings of this parameter have been significantly increased since Dobó and Csirik (2019a) with numerous novel variants.

The different modification techniques and variants, together with their formula, can be observed in Appendix A. Further, we plan to release the list of all tested settings in great detail at: https://github.com/doboandras/dsm-parameter-analysis/. Please note that some of the measures included in our analysis are not exact reproductions of the measures in the cited papers, rather they are only based on the cited measures and have been slightly adapted to be in harmony with our other measures. Further, many measures presented in this thesis are known by more than one names (e.g. Fidelity similarity is also called the sum of geometric means, Bhattacharyya coefficient and Hellinger affinity Cha (2007)). Moreover, most measures also have other variants with whom they only have very slight difference (e.g. constant multiplication, as in case of the Jensen-Shannon and Tøpsoe divergence Cha (2007)). However, these variants have the same or very similar results in most configurations, so generally it does not make much of a difference which one is used. Due to space and time limitations, it was not possible to list all names or test for all slight variants for the presented measures within this thesis, so in most such cases only the most used name and variant is reported. Furthermore, please note that a couple of measures could not be tested in case of predictive and knowledge-graph-based models due to the characteris-

tics of these models.

The following measures were selected into the basic settings set:

Cos, AdjCos, Overlap, HarmMeanMod, L_1

4.2.2 Weighting schemes (Weight; 2907)

Each word (w) is represented by a vector containing features, where each feature consists of a relation (r) and a feature word (w'). These feature vectors are created with the help of (co-occurrence) data extracted from a large corpus. However, the extracted raw frequencies alone are not completely suitable for representing the meaning of words, as most words usually occur with such common words as "is" and "the' (usually denoted as stop-words)' the most frequently, which are not really indicative of the meaning of the words (Jurafsky and Martin, 2009). Therefore it is useful to employ some weighting scheme inside the vectors to determine the strength of association between words and features, and hence the relevance of the features for the words. Similarly to vector similarity measures, they also form a very important part of DSMs, and have also been studied in many previous research (e.g. Curran, 2004; Evert, 2005; Lapesa and Evert, 2014; Kiela and Clark, 2014), as also noted in Chapter 2.

Altogether 2907 variants have been tested, which include:

- simple measures based on word-feature co-occurrence frequencies (e.g. frequecy (Curran, 2004), conditional probability (Jurafsky and Martin, 2009)),
- variants of TF/IDF and similar measures (e.g. TF/IDF (Curran, 2004) and TF/ICF (Reed et al., 2006)),

- measures based on pointwise mutual information (e.g. PMI (Church and Hanks, 1990), NPMI (Harispe et al., 2015) and LPMI (Evert, 2005)),
- other complex information theoretic and statistical measures (e.g. t-test (Curran, 2004) and odds ratio (Evert, 2005)),
- inter-rater reliability measures (e.g. Scott's pi (Scott, 1955) and Cohen's kappa (Cohen, 1960))
- and their many-many variants,
- as well as numerous new measures.

Again, more than 90% of these measures were new ones proposed by us. Our intuition for almost all of these were very similar as in case of the vector similarity measures (see Section 4.2.1), namely that we have extended some simple, conventionally used measure that already had a good performance. Again, our modifications included different weighting and transformation functions and normalization factors employed inside them, among others. The modified measures include numerous new variants of pointwise mutual information (Pmi) (Church and Hanks, 1990), conditional probability (CondProb) (Jurafsky and Martin, 2009), Rapp's measure (Rapp) (Rapp, 2003) and Lin's weighting scheme (Lin) (Lin, 1998a), among others, as well as such weighting schemes that are the combination of existing schemes.

There are also quite a few new variants combining the features of already existing versions of PMI weighting, such as the PMI_α (PmiAl) (Levy et al., 2015), the normalised PMI (NPmi) (Harispe et al., 2015), the shifted PMI (SPmi) (Weir et al., 2016), the PMI with a discounting factor (PmiWdf) (Pantel and Lin, 2002) and the

Unigram subtuples (Unis) (Pecina, 2010) measures. The number of possible settings of this parameter have been significantly increased since Dobó and Csirik (2019a) with numerous novel variants.

The different modification techniques and variants, together with their formula, can be observed in Appendix B. Further, we plan to release the list of all tested settings in great detail at: https://github.com/doboandras/dsm-parameter-analysis/. Please note that, similarly as in case of the vector similarity measures, some of the presented weighting schemes are not exact reproductions of the cited ones, rather they have been slightly adapted to be in harmony with our other schemes. Further, it was not possible to list all names or test for all slight variants for the presented measures within this thesis, as noted in case of the vector similarity measures too.

The following measures were selected into the basic settings set:

PMI, NPMI, TTest-2, OddsRatio-3, PoissonStirlingLh

4.2.3 Feature transformation techniques* (FeatTransf; 22)

Feature transformations are functions called on either the feature counts in the word vectors extracted from the corpora or on the weights of the features. They can be useful for example to reduce the skewness of feature scores (Lapesa and Evert, 2014). There were 4 major categories of settings tried, all of which were tested with several different transformation functions:

- no feature transformation (NoTransf),
- transformation of feature counts (this version could not be tested in case of predictive and knowledge-graph-based models due to the characteristics of

these models) (Freq),

- transformation of feature weights before possible smoothing and normalization (Weight BefNorm),
- and transformation of feature weights after possible smoothing and normalization (Weight AftNorm).

7 different transformation functions were tested for these. All transformation functions were designed to be interpreted on positive, zero and negative values too, and to only impact the magnitude of their argument, while keeping their sign:

$$\begin{aligned} f_{Lb}(x) &= sgn(x) \times log_2(|x| + 1) \\ f_{Sqrt}(x) &= sgn(x) \times \sqrt{|x|} \\ f_{Square}(x) &= sgn(x) \times x^2 \\ f_{Cubic}(x) &= x^3 \\ f_{Sigm}(x) &= \frac{1}{1 + e^{-x}} - 0.5 \\ f_{P1D2}(x) &= \frac{x + 1}{2} \\ f_{Rank}(x) &= valueToRank(x) \end{aligned} \tag{4.2}$$

While the base of most above functions are generally used for transformations, the idea of the P1D2 function came from Melamud et al. (2015), and that of the Rank based on Santus et al. (2016). Further, to our best knowledge, the idea of trying out the feature transformations at different steps of the DSMs (i.e. on unsmoothed frequencies, unsmoothed weights and normalized weights) is novel, the Square (Sq) and Cubic (Cu) functions have not been tried as feature

transformation functions by other authors in DSMs, and others did not define the transformation functions in such a way that they are interpreted on negative and zero values too.

Altogether 22 variants have been tested, and the following measures were selected into the basic settings set:

NoTransf, "Weight AftNorm Lb"

4.2.4 Dimensionality reduction techniques (DimRed; 21)

Dimensionality reduction can be used to reduce the number of features in the feature vectors. This can both improve the results and greatly reduce the time and space complexity of vector comparison (Landauer and Dumais, 1997; Lapesa and Evert, 2014). There were 4 major types of dimensionality reduction techniques tried, with several different dimensionality parameters in case of each:

- no dimensionality reduction (NoDimRed)
- the dimensionality reduction technique introduced by Islam and Inkpen (2008) (IslamInkpen; please note that we slightly changed the computation of the used dimensions for the vectors based on the parameter of this technique compared to Dobó and Csirik (2019a) to become fully consistent with Islam and Inkpen (2008)),
- in each vector retaining only the features with the n highest weight (inspired by the method of Islam and Inkpen (2008)) (TopNFeat),
- singular value decomposition (Landauer and Dumais, 1997; Rapp, 2003; Bullinaria and Levy, 2012) (SVD) (Please note that before the SVD, an L2

normalization was always performed, as we experienced that this step greatly enhances the results).

We have to note that when testing dimensionality reduction, we usually did a smaller number of runs than in other cases due to the rather large computational requirements of SVD and our limited resources. Further, we had to put a limit on the number of word vectors included in the SVD, and in case of Spanish we also had to set MinWFFreq to 3 instead of NoLimit when using SVD, due to the too large word-feature matrix otherwise, which would have made running SVD unmanageable. Altogether 21 variants have been tested, and the following measures were selected into the basic settings set:

NoDimRed, "IslamInkpen 0.1"

4.2.5 Smoothing techniques (Smooth; 5)

Smoothing in general can be used to reduce the noise and randomness in data, and is especially useful in case of problems with data points having zero value or probability (Jurafsky and Martin, 2009). During smoothing, the value of data points is slightly decreased in case of higher values while slightly increased in case of lower values, to reach a smoother distribution. While they are popular in many NLP applications, they have been ignored in most DSMs, with few exceptions (e.g. Dinu, 2011).

One of the most widely used group of smoothing methods in general are of the type absolute discounting (Ney and Essen, 1991), that are simple but still very powerful and efficient methods. The Kneser-Ney smoothing (KNS) (Kneser and Ney, 1995b), and its multi-discount variant, the Modified Kneser-Ney smoothing

(MKNS) (Chen and Goodman, 1999a) are widely considered to be one of the best smoothing algorithms since a long time (Chen and Goodman, 1999a; Goodman, 2001; Zhang and Chiang, 2014).

Although the probability of atomic events changes during smoothing as a necessary consequence, the marginal probabilities do not necessarily need to change, where the marginal probabilities are the probabilities obtained by summing out the probabilities of an event with respect to other events:

$$P(Y) = \sum_{z \in Z} P(Y, z) \tag{4.3}$$

One of the key motivations when developing the KNS was that it should preserve the marginal distributions of the original model, meaning that the obtained model satisfies the following equation:

$$\frac{c(w_i)}{\sum_{w_i} c(w_i)} = \sum_{w_{i-1}} p(w_i|w_{i-1})\, p(w_{i-1}) \tag{4.4}$$

This is very advantageous in many cases, and under certain assumptions, an optimal model can only be obtained by satisfying this property, as discussed by Goodman in the extended version of his paper (Goodman, 2001). Hence Goodman comes to the conclusion that under these assumptions any smoothing method not preserving the original marginals can be improved by modifying it to preserve them. Despite this fact, many frequently used smoothing techniques, including the MKNS, do not satisfy this property: when Chen and Goodman (1999a) refined the original KNS by introducing three discount parameters instead of just one, they did not adjust the lower-order distributions according to this change, which resulted in the loss of the original marginals in the smoothed

model.

Therefore I have devised such a novel smoothing method based on the MKNS, that keeps all the advantages of both the KNS and the MKNS, while also preserving the original marginal distributions (Dobó, 2018). The final form for this new smoothing technique, called Multi-D Kneser-Ney Smoothing Preserving the Original Marginal Distributions (MDKNSPOMD), for a bigram language model is as follows:

$$\begin{aligned} p_{MDKNSPOMD}\left(w_i|w_{i-1}\right) =& \frac{c\left(w_{i-1}w_i\right) - D\left(c\left(w_{i-1}w_i\right)\right)}{\sum_{w_i} c\left(w_{i-1}w_i\right)} + \\ & \gamma_{MDKNSPOMD}\left(w_{i-1}\right) p_{MDKNSPOMD}\left(w_i\right) \end{aligned} \tag{4.5}$$

$$\begin{aligned} \gamma_{MDKNSPOMD}\left(w_{i-1}\right) =& \frac{D_1 N_1\left(w_{i-1}.\right) + D_2 N_2\left(w_{i-1}.\right)}{\sum_{w_i} c\left(w_{i-1}w_i\right)} + \\ & \frac{D_{3+} N_{3+}\left(w_{i-1}.\right)}{\sum_{w_i} c\left(w_{i-1}w_i\right)} \end{aligned} \tag{4.6}$$

$$p_{MDKNSPOMD}\left(w_i\right) = \frac{D_1 N_1\left(.w_i\right) + D_2 N_2\left(.w_i\right) + D_{3+} N_{3+}\left(.w_i\right)}{D_1 N_1\left(..\right) + D_2 N_2\left(..\right) + D_{3+} N_{3+}\left(..\right)} \tag{4.7}$$

We have to note that when testing the various smoothing options, we usually did a smaller number of runs than in other cases due to the rather large computational requirements of smoothing and our limited resources. Altogether 5 variants of smoothing techniques have been tried:

- no smoothing (NoSmooth)
- Kneser-Ney smoothing (Kneser and Ney, 1995a) on weights (Weight KNS),
- Kneser-Ney smoothing (Kneser and Ney, 1995a) on raw counts (Freq KNS),

- Modified Kneser-Ney smoothing (Chen and Goodman, 1999b) on raw counts (Freq MKNS),
- Multi-D Kneser-Ney Smoothing Preserving the Original Marginal Distributions (Dobó, 2018) on raw counts (Freq MDKNSPOMD),

To our best knowledge none of the smoothing variants used in our analysis have ever been tried in DSMs, and we are also the first to try smoothing at multiple points in DSMs. The smoothing parameters used in all versions are the estimates of the optimal parameters determined by the method described in Chen and Goodman (1999b), calculated on the BNC. The following measures were selected into the basic settings set:

NoSmooth

4.2.6 Vector normalization methods* (VecNorm; 3)

Although there are such distance and similarity measures that are independent of vector magnitudes (e.g. cosine similarity), most measures are not so. Therefore normalizing the vectors before comparing them makes sense. This aspect of DSMs has also only been considered in few studies (e.g. Jones and Furnas, 1987; Yin and Schütze, 2016).

The most common way for normalization is by their L_2 norm. However, as it will be seen, there are many similarity measures originally developed for comparing probability distributions. These measures assume such vectors as input, whose values are non-negative (or even positive) and sum up to 1. To be as consistent with the theoretical background of these measures as possible, we have also evaluated the L_1 normalization of the vectors.

In case of the word vectors provided by Mikolov et al. (2013b), the vectors were already L_2 normalized, so there the unnormalized version (NN) could not be tried. Further, please note that in many cases the feature vectors also include negative and zero weights (e.g. due to the weighting scheme used, and also in case of the Mv (Mikolov et al., 2013b)), so despite using the L_1 normalization, their elements very rarely sum up to 1, and thus they are still not fully adequate for measures coming from probability theory.

Altogether the above 3 variants (NN, L_1, L_2) have been tested, and the following measures were selected into the basic settings set:

L_2, L_1

4.2.7 Filtering stop-words (StopW; 2)

Stop-words are such very frequently used words, whose usage as context words are very uninformative and not useful in most cases (as also noted in Section 4.2.2). Therefore stop-words have usually been filtered not just in DSMs, but also in many other NLP applications since a very long time (Manning and Schütze, 1999). While they can prove to be very useful (Huang et al., 2012), others conclude that removing these words in DSMs does not improve performance (Bullinaria and Levy, 2012), probably due to the used weighting schemes already assigning a very low weight to them, essentially already filtering them out almost completely (Kiela and Clark, 2014).

Both possibilities (True, False) have been tested, using the Stopwords ISO collection[1], with the following measures in the basic settings set:

[1] https://github.com/stopwords-iso/

False

4.2.8 Minimum limits on word-feature tuple frequencies (MinWFFreq; 6)

As pointwise mutual information (PMI) becomes unstable in case of small co-occurrence frequencies, it is better to only consider such word-feature pairs whose frequency is above a given threshold in case of this weighting scheme (Church and Hanks, 1990). Based on these findings, this feature might also be useful in case of other weighting schemes too, so it seemed interesting to test this as a general option, and not just in case of PMI weighting.

Altogether 6 variants have been tested, and the following measures were selected into the basic settings set:

NoLimit

4.2.9 Minimum limits on word-feature tuple weights* (MinWFWeight; 26)

Negative PMI values can also be unreliable, and thus several researchers suggest to discard these (Dagan et al., 1995; Bullinaria and Levy, 2007). Similarly to the minimum limit on word-feature tuple frequencies, this option might be useful in case of other weighting schemes too, so we also tested this generally, not just in case of PMI weighting.

We have tried two variants. In the first version (Limit) a weight is replaced with the limit if it is below it:

$$Limit(w, minValue) = \begin{cases} w & if\ w \geq minValue \\ minValue & otherwise \end{cases} \tag{4.8}$$

In the second version (Zero) a weight is replaced with zero if it is below the limit:

$$Zero(w, minValue) = \begin{cases} w & if\ w \geq minValue \\ 0 & otherwise \end{cases} \tag{4.9}$$

Several limit values have been tested for both versions. Previous studies usually used either the NoLimit option or an option that is equivalent to our "Zero 0" (and "Limit 0") version. Our motivation for testing negative parameters with the Limit version was that the "Zero 0" option seems to be a bit too strict, and results in the same zero score for both an original score of zero and an original score with negative sign and large magnitude. The Limit variant with a negative parameter does almost the same, but keeps the sign of negative values while restricting their magnitude. On the other hand, the Zero version with a positive parameter filters out the unimportant features with low weight for each word, essentially doing something similar to dimensionality reduction or stop word filtering. It seemed logical to also test parameter values with the opposite sign in case of both versions, and with several different magnitudes of the parameter. To our best knowledge, no one has ever tested the Limit or Zero options in DSMs, nor thresholds other than 0.

The number of possible settings of this parameter have been significantly increased since Dobó and Csirik (2019a) with numerous novel variants. The following measures were selected into the basic settings set:

NoLimit, "Zero 0"

4.2.10 Minimum limits on feature frequencies (MinFFreq; 14)

Based on the ideas of the above two parameters, we have thought that it could be interesting to also test whether discarding features that are very infrequent on the whole (having a total frequency of or below a given limit) would improve the results of DSMs or not. Although mainly due to computational efficiency reasons, this technique has already been employed by others too (e.g. Levy et al., 2015).

Altogether 14 variants have been tested, and the following measures were selected into the basic settings set:

NoLimit

CHAPTER 5

Semantic similarity of English words

5.1 The first phase of the heuristic approach

In this phase the most promising parameter settings had to be selected for each parameter based on multiple runs for each setting on the MD1 development dataset. In the following subsections, the detailed performance of the different settings are presented and evaluated for each parameter. The settings for the second phase are selected based on the results achieved during the multiple runs of the settings, using the 9 measures presented in Section 3.2. As already mentioned before, the number of possible settings of several parameters have been significantly increased since Dobó and Csirik (2019a) with numerous novel variants.

Singular value decomposition for dimensionality reduction has very large

time and space complexity, therefore, due to our limited resources, when testing SVD in the second phase using the DcBnc, only a reduced settings set was used in case of each parameter.

In case of each figure presenting the results, those measures marked with an * were selected to be included in the second phase, with those marked with ** also being part of the reduced settings set. In case of those parameters, where a very large number of settings were tested for, the results for only a small proportion of settings can be shown here due to space limitations, but the full results are planned to be made publicly available at:

https://github.com/doboandras/dsm-parameter-analysis/.

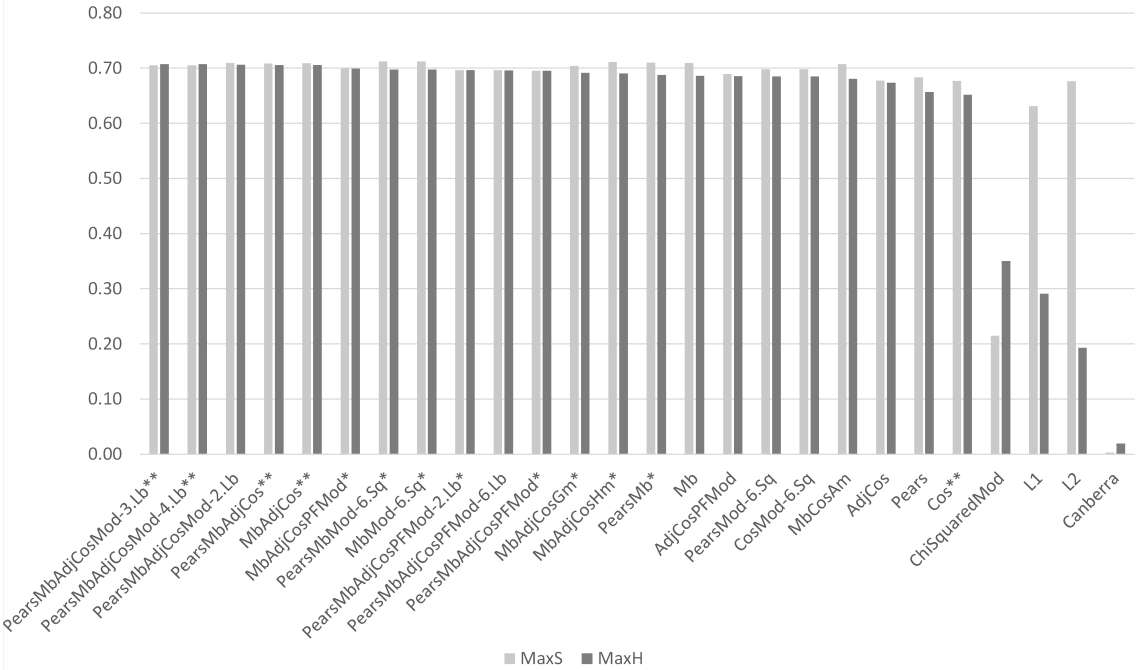
5.1.1 Results using the counts of Dobó and Csirik (2013) on the BNC

5.1.1.1 Vector similarity measures

Analyzing the results (see Figure 5.1) it can be seen that most measures based on the inner product, also including variants of the cosine similarity, correlation measures and statistical coefficients, as well as measures proposed by Lin (Lin, 1998a), and the variants of these, generally performed well. On the other hand, distance-based measures, such as the Minkowski distances (L_p) or the Canberra distance, achieved relatively low H scores, mostly due to their low Pearson correlation scores. Measures designed for comparing probability distributions and for statistical hypothesis testing mostly also performed poorly. A large proportion of the best measures are combinations of multiple measures and modified variants of some existing measures, proposed by us.

The best measures achieved an H score above 0.7, from which altogether 13 (out of 1221) measures were selected to the second phase, and 5 were also selected into the reduced settings set.

Figure 5.1. First-phase performance of vector similarity measures using the DcBnc.



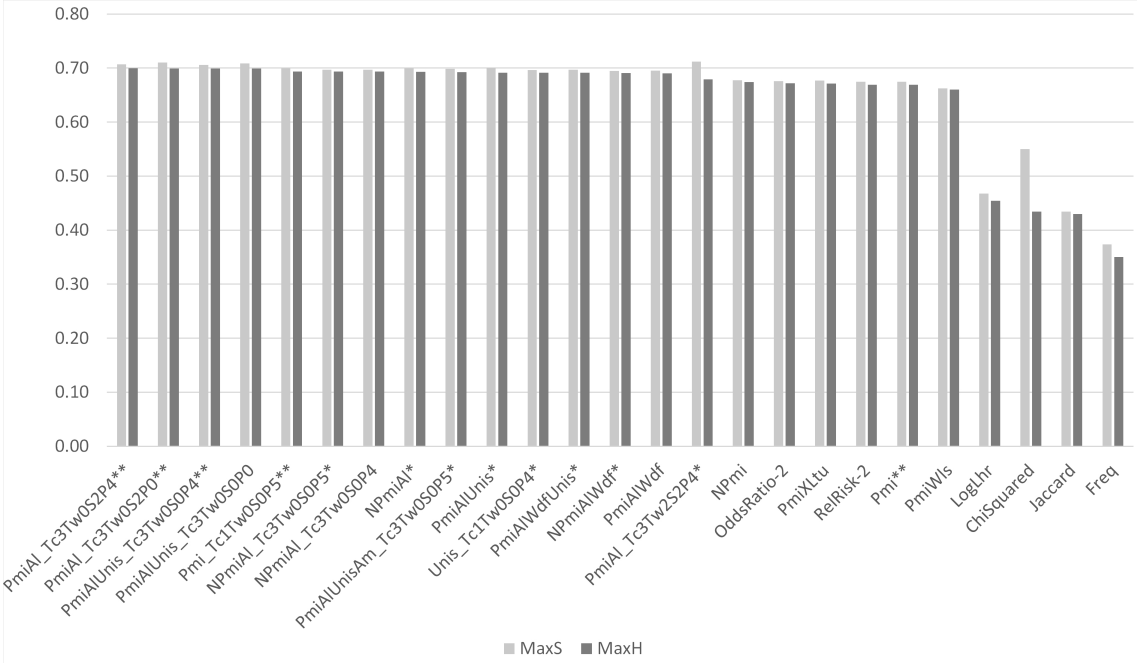
5.1.1.2 Weighting schemes

The best results (see Figure 5.2) were clearly achieved by variants of the pointwise mutual information. Beside these, some other complex information theoretic and statistical measures also scored high, while inter-rater reliability measures generally performed a little worse. Simple measures based on word-feature co-occurrence frequencies generally achieved relatively low H scores. A large proportion of the best measures are new ones proposed by us.

Similarly as in case of the vector similarity measures, the highest H scores are

close to 0.7. Altogether 13 (out of 2907) measures were selected to the second phase, from which 5 were also selected into the reduced settings set.

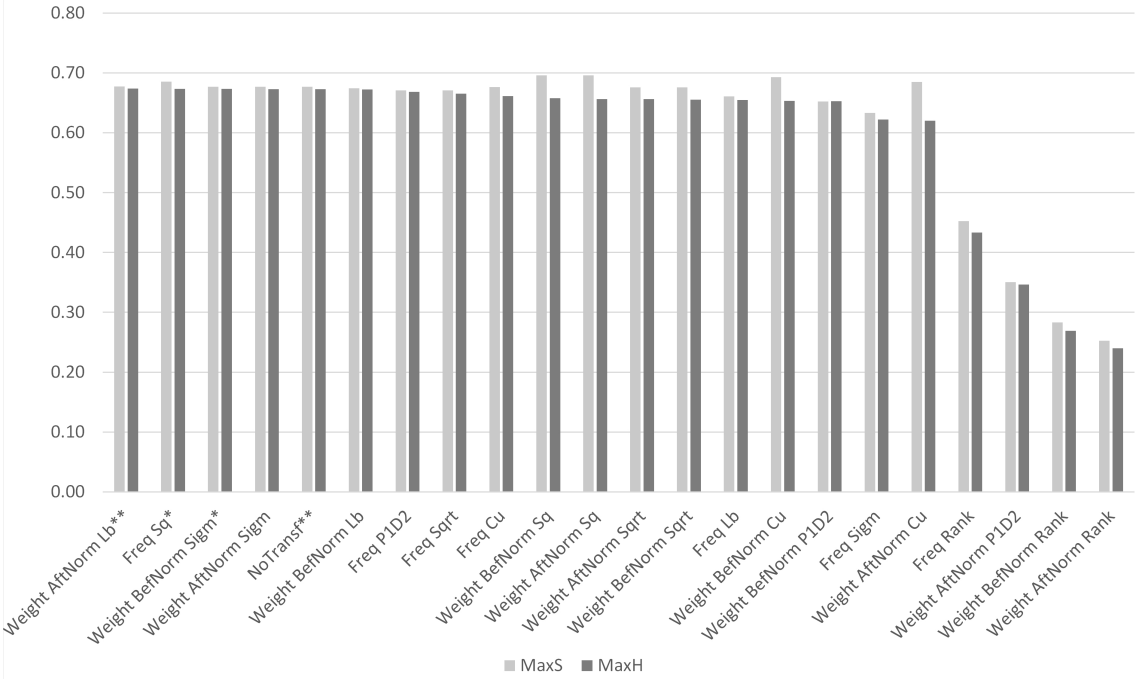
Figure 5.2. First-phase performance of weighting schemes using the DcBnc.



5.1.1.3 Feature transformation

The results (see Figure 5.3) are rather mixed in case of this parameter. Settings with all 4 major feature transformation categories have achieved good results, although with different transformation functions in case of each. Further, a clear ranking cannot be determined in case of the different transformation functions based on their results either. Altogether 4 (out of 22) settings were selected to the second phase, from which 2 were also selected into the reduced settings set.

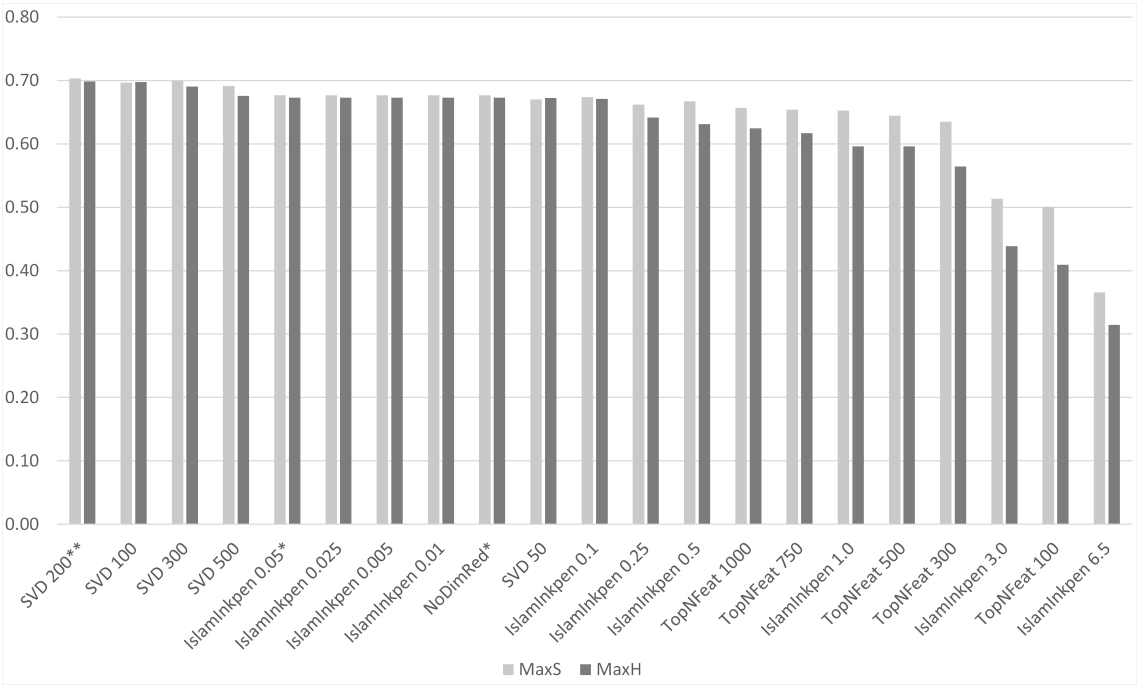
Figure 5.3. First-phase performance of feature transformation techniques using the DcBnc.



5.1.1.4 Dimensionality reduction

The results (see Figure 5.4) show that the best results were achieved by singular value decomposition with different dimensionality parameters, followed by the technique of Islam and Inkpen (2008). Please note that we slightly changed the computation of the used dimensions for the vectors based on the parameter of this technique compared to Dobó and Csirik (2019a) to become fully consistent with Islam and Inkpen (2008). Altogether 3 (out of 21) settings were selected to the second phase (2 for the normal set and 1 for the reduced set).

Figure 5.4. First-phase performance of dimensionality reduction techniques using the DcBnc.



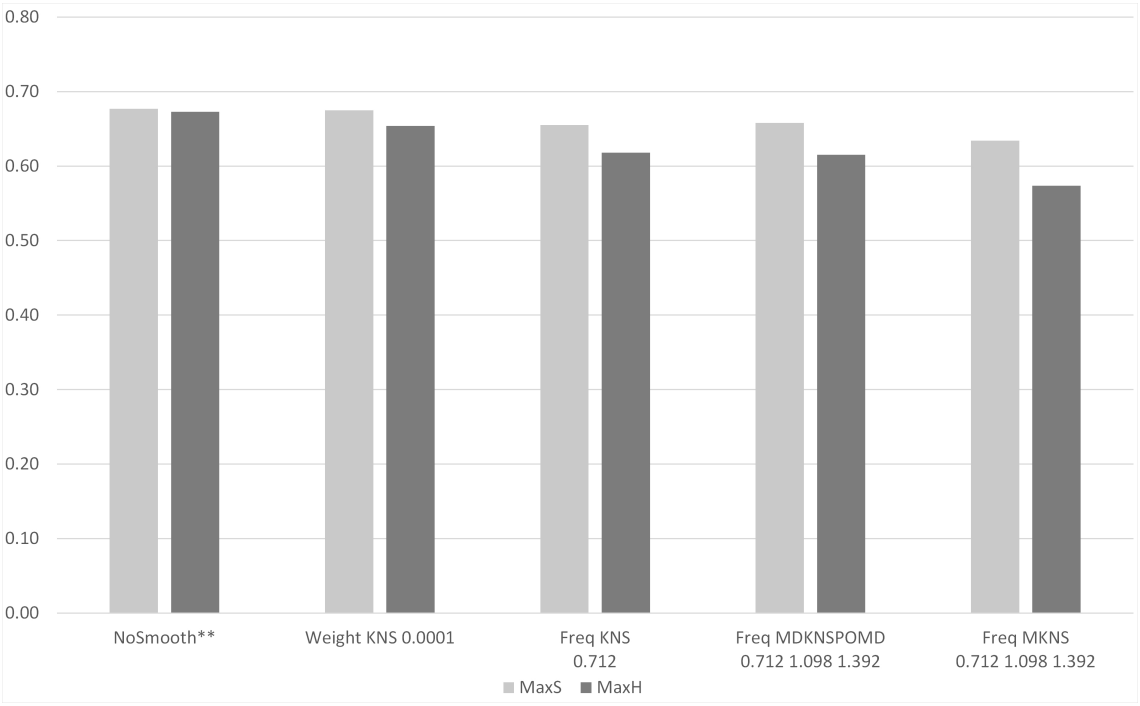
5.1.1.5 Smoothing

Smoothing seems to considerably worsen the results of DSMs (see Figure 5.5), while posing significant extra time and space complexity burden on them. Therefore only the no smoothing setting (out of 5) was selected for the next phase (also included in the reduced settings set).

5.1.1.6 Vector normalization

As many vector similarity measures are independent of vector normalization, many configurations achieve the same results irrespective of which normalization technique is used. The best configuration from the basic set is also achieved by a measure independent of vector normalization, therefore the best scores of all 3

Figure 5.5. First-phase performance of smoothing techniques using the DcBnc.

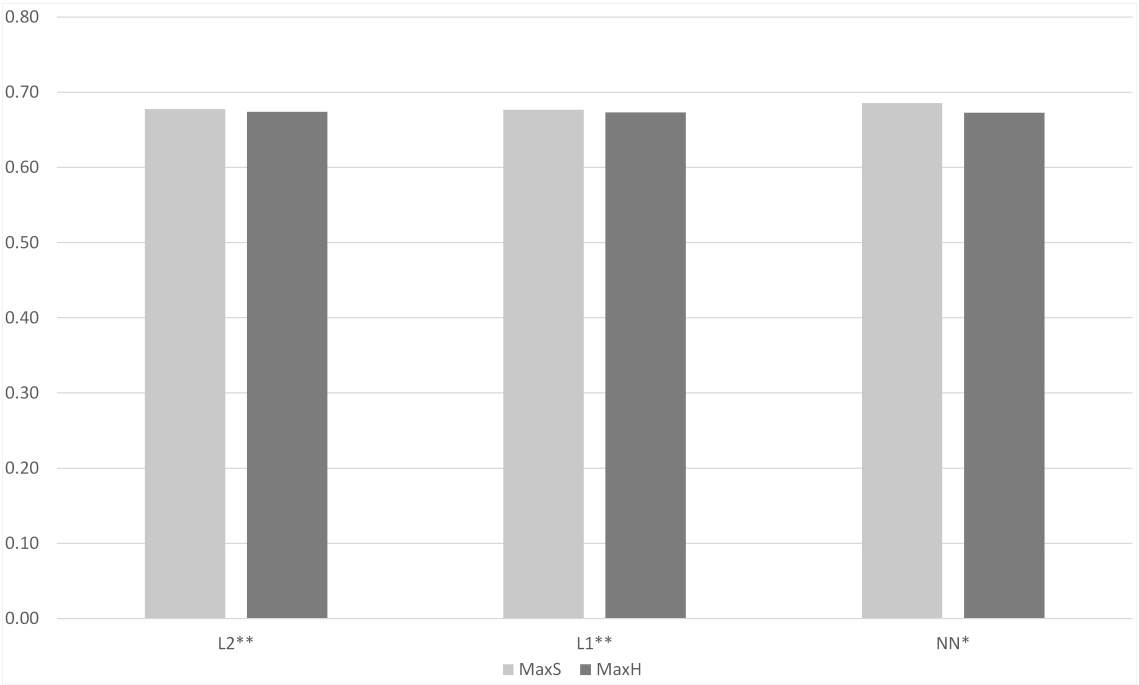


techniques are the same (see Figure 5.6). All of them were selected for the next phase, with the no normalization option omitted from the reduced settings set.

5.1.1.7 Filtering stop-words

As both filtering and not filtering stop-words seem to achieve good results (see Figure 5.7), both options were selected for the next phase, with only the False option included in the reduced settings set. The reason why filtering or not filtering these words does not have a huge impact on performance is probably due to the fact that using a proper weighting scheme already devaluates these words so much as if they were almost completely filtered out (Kiela and Clark, 2014), as already noted before.

Figure 5.6. First-phase performance of vector normalization techniques using the DcBnc.



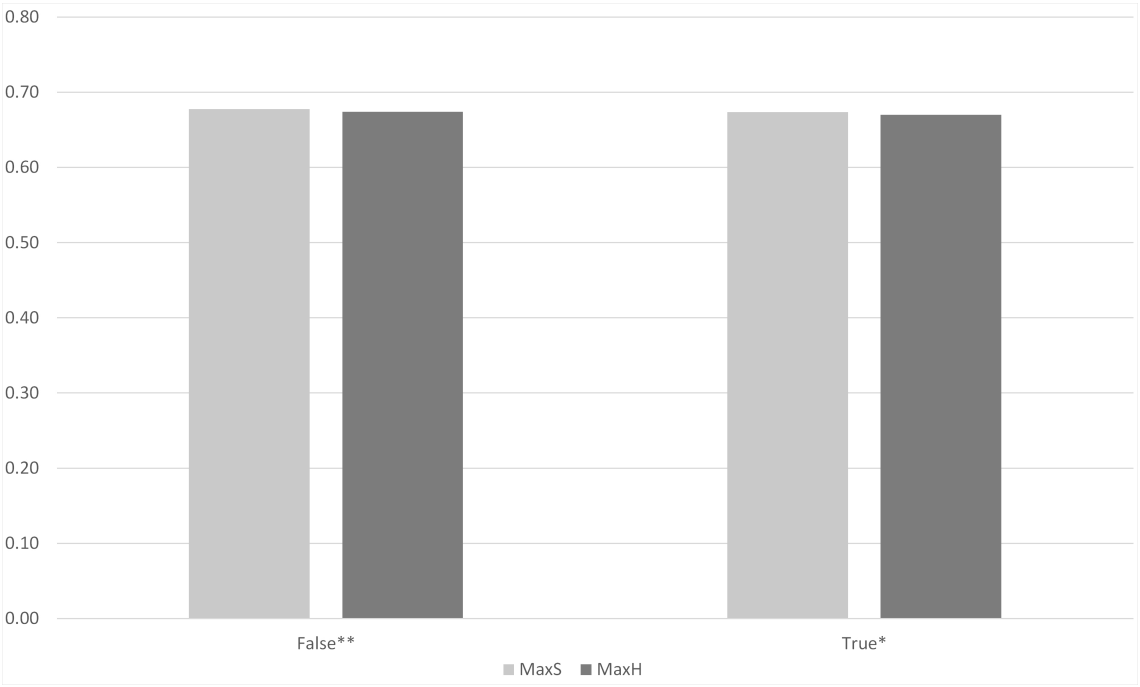
5.1.1.8 Minimum limits on word-feature tuple frequencies

As employing minimum limits on word-feature tuple frequencies seems to deteriorate performance (see Figure 5.8), only the no limit option (out of 6) was selected for the second phase (also included in the reduced settings set).

5.1.1.9 Minimum limits on word-feature tuple weights

The results for word-feature tuple weights are somewhat mixed, but the Zero option having a small positive parameter seems to be slightly superior to the other settings (see Figure 5.9). Altogether 5 (out of 26) settings were selected to the second phase, from which 3 were also selected into the reduced settings set.

Figure 5.7. First-phase performance achieved by filtering and not filtering stop-words using the DcBnc.



5.1.1.10 Minimum limits on feature frequencies

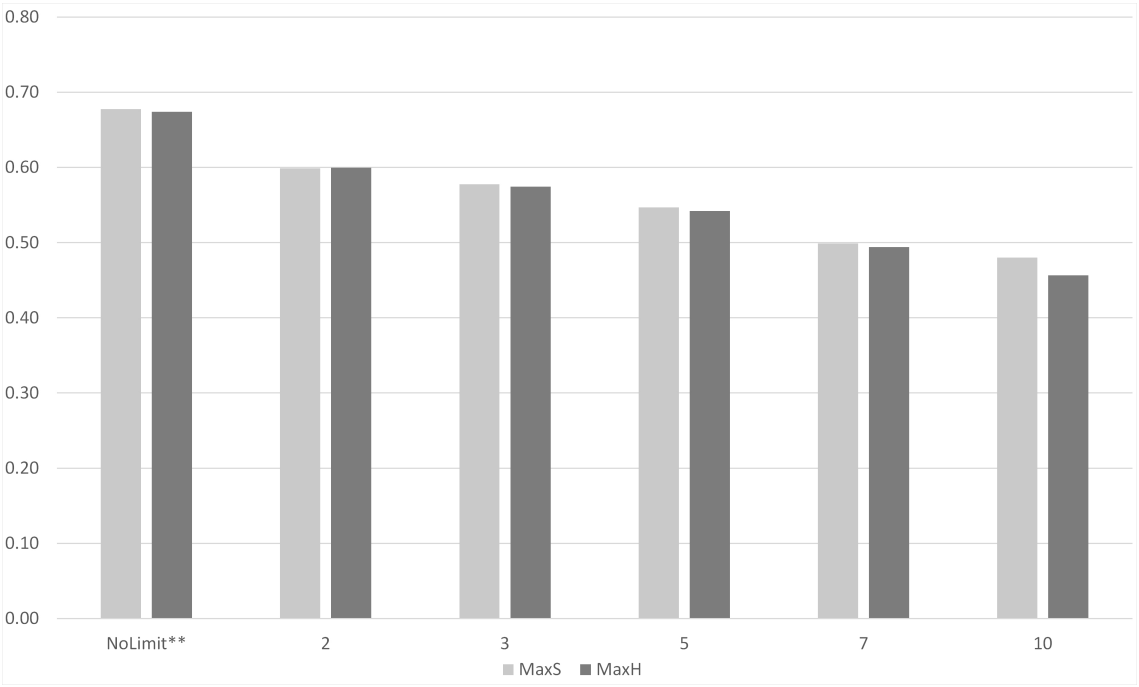
As setting no limit on feature frequencies seems to achieve the best results (see Figure 5.10), only this option (out of 14) was selected for the next phase (also included in the reduced settings set).

5.1.2 Results using the semantic vectors of Mikolov et al. (2013b)

5.1.2.1 Vector similarity measures

These measures show very similar results using the Mv (see Figure 5.11), as using the DcBnc (see Figure 5.1). Measures based on the inner product, also including variants of the cosine similarity, correlation measures and statistical coefficients, achieved the highest H scores. Further, distance-based measures and measures

Figure 5.8. First-phase performance achieved by setting minimum limits on word-feature tuple frequencies using the DcBnc.



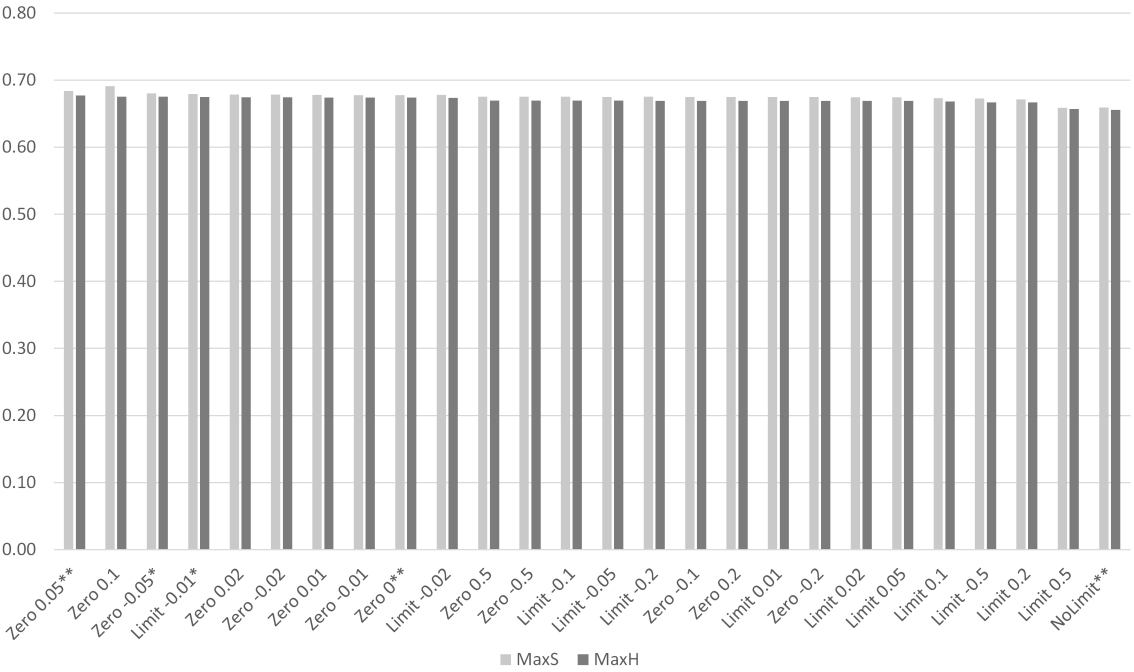
designed for comparing probability distributions and for statistical hypothesis testing performed poorly most of the time. Again, some of our newly proposed measures achieved the highest scores.

The best measures achieved an H score around 0.75, from which altogether 16 (out of 1221) measures were selected for the second phase.

5.1.2.2 Feature transformation

The results (see Figure 5.12) are rather mixed in case of this parameter using the Mv too, but resemble the results using the DcBnc more or less (see Figure 5.3). Altogether 6 (out of 15, as some of the settings could not be tested with this corpus (see Section 4.2)) settings were selected to the second phase.

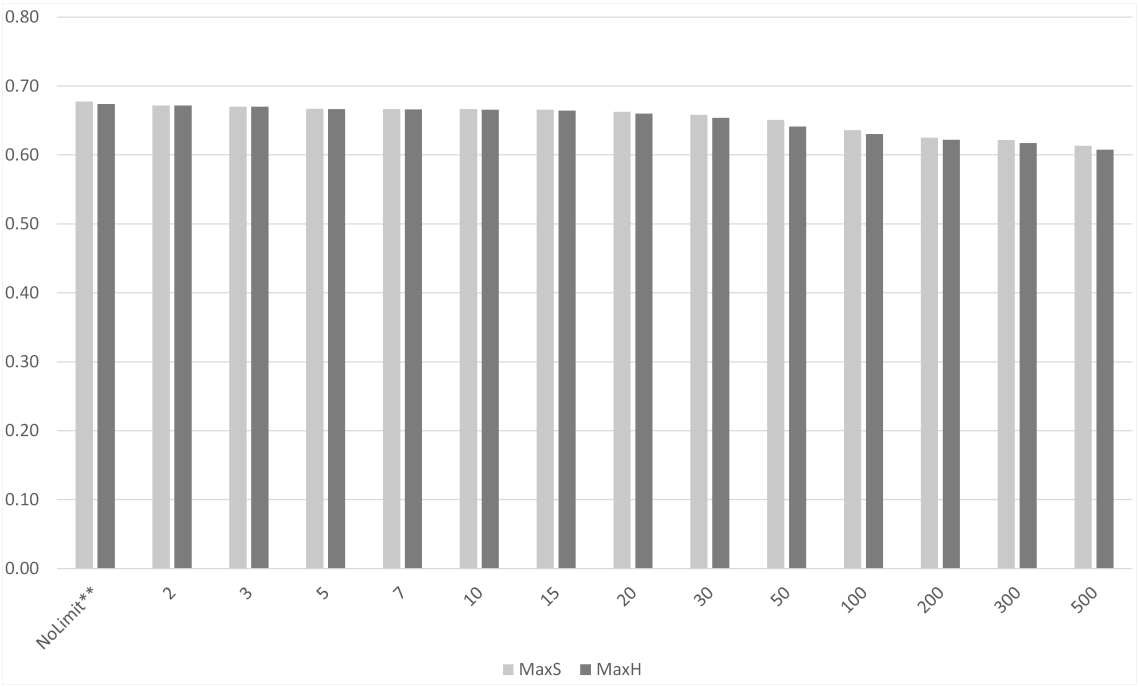
Figure 5.9. First-phase performance achieved by setting minimum limits on word-feature tuple weights using the DcBnc.



5.1.2.3 Vector normalization

As noted before, many vector similarity measures are independent of vector normalization, so they achieve the same results irrespective of which normalization technique is used. The best configuration from the basic set is achieved by such a measure here too, similarly as it was in case of the DcBnc, therefore the best scores of both techniques (the NN option could not be tested with the vectors of Mikolov et al. (2013b) (see Section 4.2)) are the same (see Figure 5.13). Both of them were selected for the next phase.

Figure 5.10. First-phase performance achieved by the setting minimum limits on feature frequencies using the DcBnc.



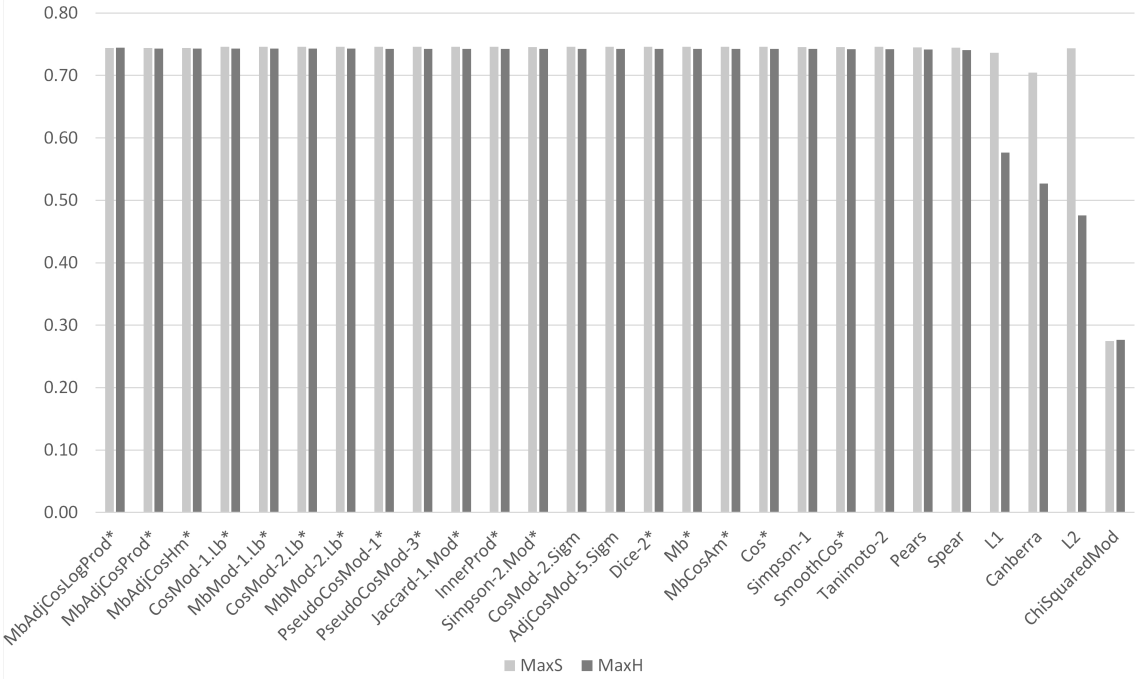
5.1.2.4 Minimum limits on word-feature tuple weights

Setting no limit or a negative limit with larger magnitude in case of either the Zero or the Limit option achieved the best results in case of word-feature tuple weights by far (see Figure 5.14). Altogether 8 (out of 26) settings were selected to the second phase.

5.2 The second phase of the heuristic approach

The purpose of this phase was to determine what the best possible configuration is in case of count-vector-based (using the DcBnc) and predictive (using the Mv) DSMs by testing all combinations of the selected settings for all parameters. A

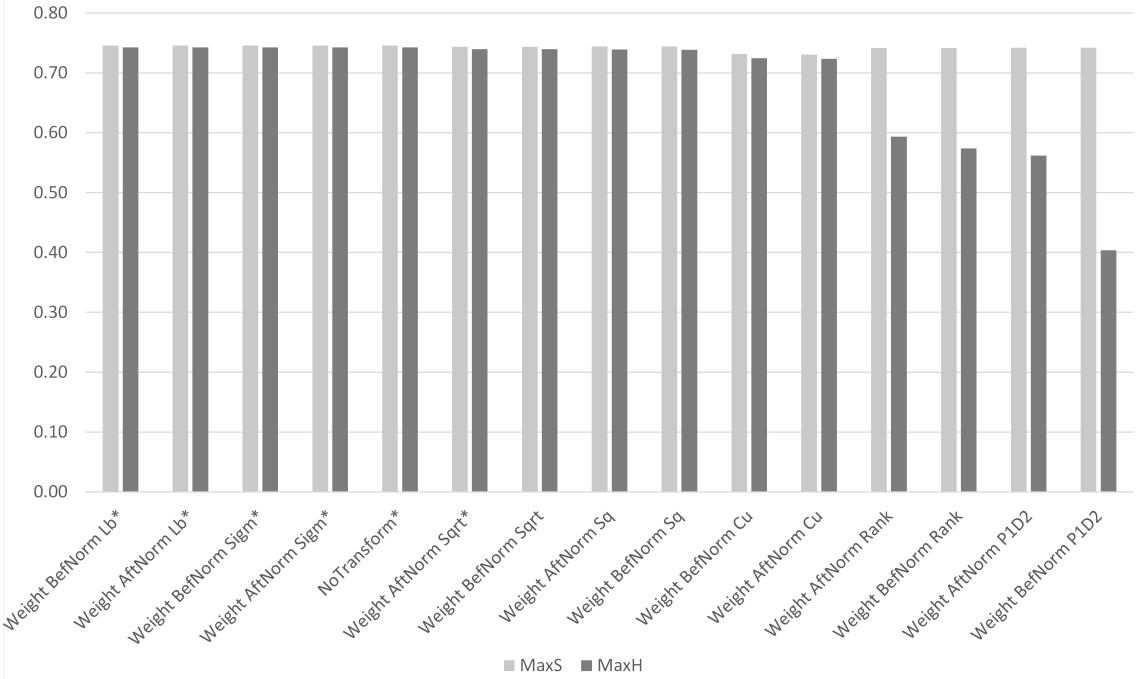
Figure 5.11. First-phase performance of vector similarity measures using the Mv.



dataset distinct from the one used in the first phase, namely the MD2 development dataset, was used for evaluation.

As a full test of all possible configurations would have been unfeasible in case of count-vector-based DSMs (see Section 4.1), only a heuristic analysis with the parameter settings selected in the first phase was performed. In case of predictive DSMs, a full analysis was also feasible in Dobó and Csirik (2019a), when the number of tested settings for these 4 parameters were still considerably lower than now. Therefore beside the heuristic analysis, a full analysis was also done at that time to validate the results (see Section 5.2.3).

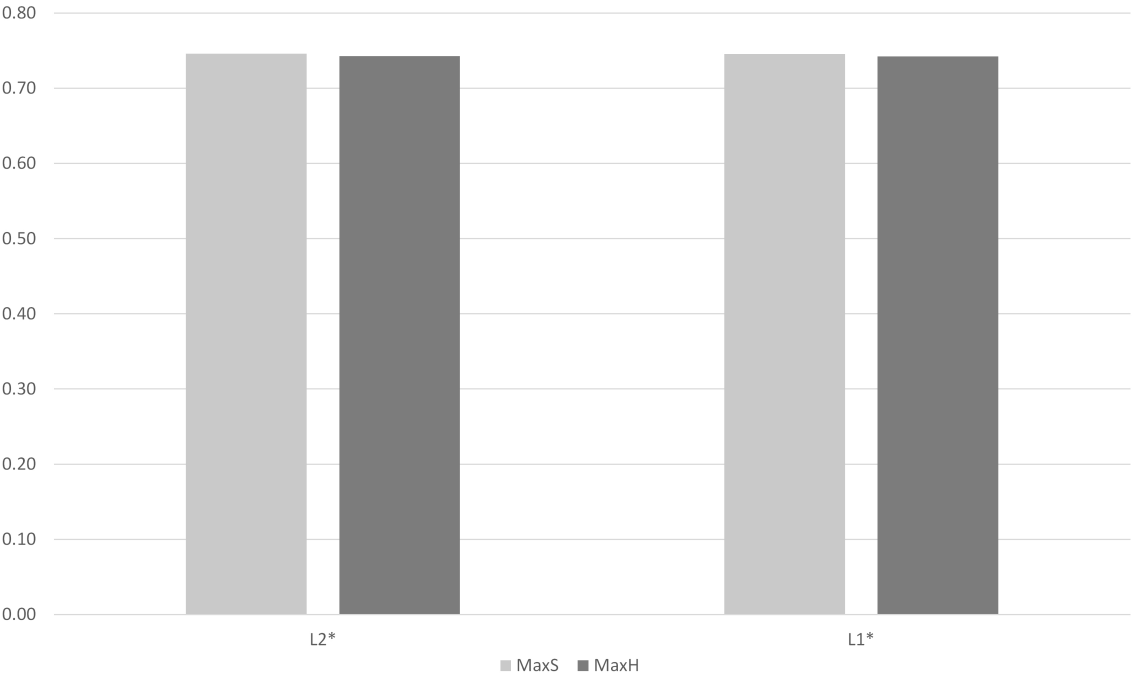
Figure 5.12. First-phase performance of feature transformation techniques using the Mv.



5.2.1 Results using the counts of Dobó and Csirik (2013) on the BNC

With the settings of the 10 parameters selected in the first phase, altogether 40860 configurations were tested (in case of the singular value decomposition settings for dimensionality reduction, only a reduced set of settings were used, as already noted in Section 4.2). A very small proportion of these, together with their performance, are presented in Table 5.1. The configuration with the best results is noted as BestCvbmDcBnc2. We have to note that there were actually two distinct configurations with the same best score, and they were only different in their DimRed parameter setting. We have chosen the one with the "IslamInkpen 0.05" setting as best (BestCvbmDcBnc2), as that setting achieved better performance in the first phase than the "NoDimRed" setting in the other configuration.

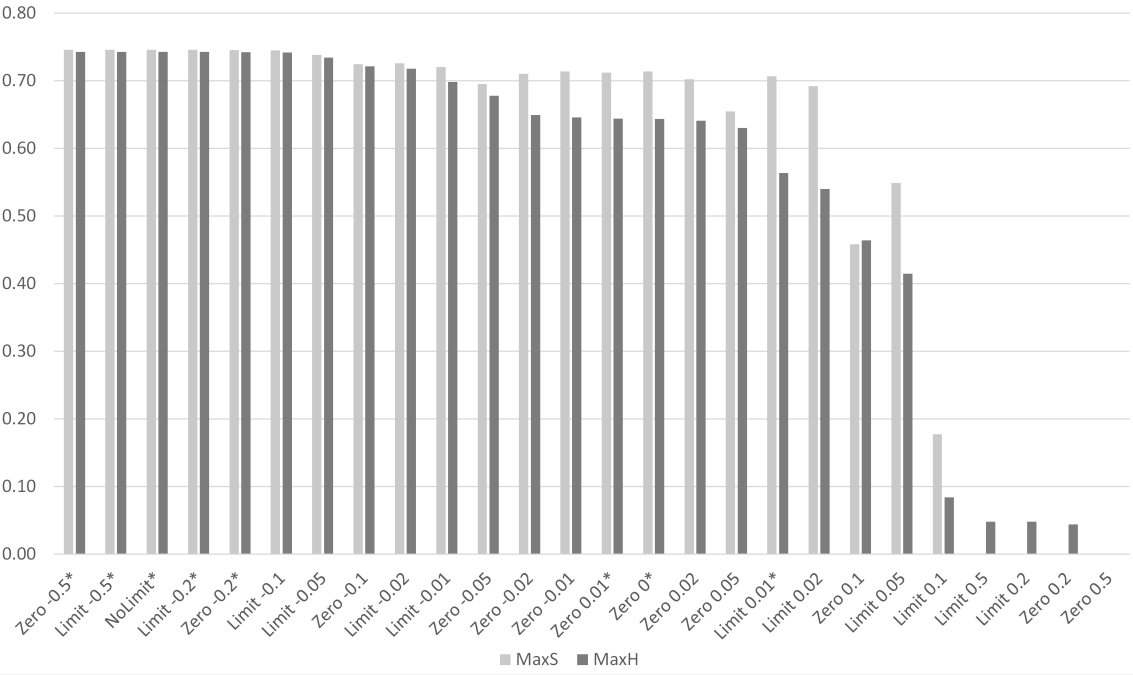
Figure 5.13. First-phase performance of vector normalization techniques using the Mv.



As already mentioned before, the number of possible settings of several parameters have been significantly increased since Dobó and Csirik (2019a) with numerous novel variants. The results of the best found count-vector-based configuration with those reduced number of settings for several parameters will also be included in the rest of the thesis (BestCvbmDcBnc).

Further we also wanted to include the results achieved with the selected best configuration in case of predictive DSMs (see Table 5.2), run on the DcBnc, in Table 5.1. However, as only 4 out of 10 parameters could be used in case of predictive DSMs, the hypothetical best configuration of the 10 parameters were produced by changing the settings of these 4 parameters in the best method on count-vector-based DSMs to the settings of the best method on predictive DSMs, leaving the other 6 parameter settings unchanged (as they were in the best con-

Figure 5.14. First-phase performance achieved by setting minimum limits on word-feature tuple weights using the Mv.



figuration on count-vector-based DSMs). This hypothetical best configuration is noted as BestPmMv2OnCvbm.

Although presenting the definition of all settings for every parameter would be impossible within this thesis due to their large number, as also noted before, below we define a couple of them to help interpreting our most important results.

The vector similarity measures used in BestCvbmDcBnc2 is a combination of the Pearson (Pears) (Jones and Furnas, 1987), Maryland Bridge (Mb) Deza and Deza (2016) and Adjusted cosine (AdjCos) Shalaby and Zadrozny (2016) measures, with some additional transformations:

Table 5.1. Second-phase performance of a selection of configurations using the DcBnc.

Abbrev	Parameter settings							P	S	H
BestCvbmDcBnc2	**VecSim**		**Weight**			**FeatTransf**		**0.72**	**0.71**	**0.71**
	PearsMbAdjCosMod-3.Lb		PmiAl-Tc3Tw0S2P0			NoTransf				
	DimRed	**Smooth**	**VecNorm**	**StopW**	**MinWFFreq**	**MinWFWeight**	**MinFFreq**			
	IslamInkpen 0.05	NoSmooth	L_1	false	NoLimit	Zero 0	NoLimit			
-	**VecSim**		**Weight**			**FeatTransf**		**0.72**	**0.71**	**0.71**
	PearsMbAdjCosMod-3.Lb		PmiAl-Tc3Tw0S2P0			NoTransf				
	DimRed	**Smooth**	**VecNorm**	**StopW**	**MinWFFreq**	**MinWFWeight**	**MinFFreq**			
	NoDimRed	NoSmooth	L_1	false	NoLimit	Zero 0	NoLimit			
-	**VecSim**		**Weight**			**FeatTransf**		**0.72**	**0.71**	**0.71**
	PearsMbAdjCosMod-3.Lb		PmiAl-Tc3Tw0S2P0			NoTransf				
	DimRed	**Smooth**	**VecNorm**	**StopW**	**MinWFFreq**	**MinWFWeight**	**MinFFreq**			
	NoDimRed	NoSmooth	L_1	false	NoLimit	Zero -0.05	NoLimit			
-	**VecSim**		**Weight**			**FeatTransf**		**0.72**	**0.71**	**0.71**
	PearsMbAdjCosMod-3.Lb		PmiAl-Tc3Tw0S2P0			NoTransf				
	DimRed	**Smooth**	**VecNorm**	**StopW**	**MinWFFreq**	**MinWFWeight**	**MinFFreq**			
	IslamInkpen 0.05	NoSmooth	L_1	false	NoLimit	Zero -0.05	NoLimit			
-	**VecSim**		**Weight**			**FeatTransf**		**0.72**	**0.71**	**0.71**
	PearsMbAdjCosMod-4.Lb		PmiAl-Tc3Tw0S2P0			NoTransf				
	DimRed	**Smooth**	**VecNorm**	**StopW**	**MinWFFreq**	**MinWFWeight**	**MinFFreq**			
	NoDimRed	NoSmooth	L_1	false	NoLimit	Zero -0.05	NoLimit			
-	**VecSim**		**Weight**			**FeatTransf**		**0.72**	**0.71**	**0.71**
	PearsMbAdjCosMod-4.Lb		PmiAl-Tc3Tw0S2P0			NoTransf				
	DimRed	**Smooth**	**VecNorm**	**StopW**	**MinWFFreq**	**MinWFWeight**	**MinFFreq**			
	IslamInkpen 0.05	NoSmooth	L_1	false	NoLimit	Zero 0	NoLimit			
-	**VecSim**		**Weight**			**FeatTransf**		**0.72**	**0.71**	**0.71**
	PearsMbAdjCosMod-4.Lb		PmiAl-Tc3Tw0S2P0			NoTransf				
	DimRed	**Smooth**	**VecNorm**	**StopW**	**MinWFFreq**	**MinWFWeight**	**MinFFreq**			
	NoDimRed	NoSmooth	L_1	false	NoLimit	Zero 0	NoLimit			
-	**VecSim**		**Weight**			**FeatTransf**		**0.72**	**0.71**	**0.71**
	PearsMbAdjCosMod-3.Lb		PmiAl-Tc3Tw0S2P0			NoTransf				
	DimRed	**Smooth**	**VecNorm**	**StopW**	**MinWFFreq**	**MinWFWeight**	**MinFFreq**			
	IslamInkpen 0.05	NoSmooth	L_1	false	NoLimit	Zero 0.05	NoLimit			
-	**VecSim**		**Weight**			**FeatTransf**		**0.72**	**0.71**	**0.71**
	PearsMbAdjCosMod-3.Lb		PmiAl-Tc3Tw0S2P0			NoTransf				
	DimRed	**Smooth**	**VecNorm**	**StopW**	**MinWFFreq**	**MinWFWeight**	**MinFFreq**			
	NoDimRed	NoSmooth	L_1	false	NoLimit	Zero 0.05	NoLimit			
-	**VecSim**		**Weight**			**FeatTransf**		**0.72**	**0.71**	**0.71**
	PearsMbAdjCosMod-4.Lb		PmiAl-Tc3Tw0S2P0			NoTransf				
	DimRed	**Smooth**	**VecNorm**	**StopW**	**MinWFFreq**	**MinWFWeight**	**MinFFreq**			
	IslamInkpen 0.05	NoSmooth	L_1	false	NoLimit	Zero -0.05	NoLimit			
-	**VecSim**		**Weight**			**FeatTransf**		**0.68**	**0.70**	**0.69**
	Cos		PmiAl-Tc3Tw0S2P0			Weight AftNorm Lb				
	DimRed	**Smooth**	**VecNorm**	**StopW**	**MinWFFreq**	**MinWFWeight**	**MinFFreq**			
	SVD 200	NoSmooth	L_2	false	NoLimit	Zero 0	NoLimit			
BestCvbmDcBnc	**VecSim**		**Weight**			**FeatTransf**		**0.66**	**0.69**	**0.68**
	Cos		WPmi-7			Weight AftNorm Lb				
	DimRed	**Smooth**	**VecNorm**	**StopW**	**MinWFFreq**	**MinWFWeight**	**MinFFreq**			
	SVD 200	NoSmooth	L_2	false	NoLimit	Zero 0	NoLimit			
BestPmMv2OnCvbm	**VecSim**		**Weight**			**FeatTransf**		**0.50**	**0.65**	**0.57**
	MbAdjCosLogProd		PmiAl-Tc3Tw0S2P0			Weight BefNorm Lb				
	DimRed	**Smooth**	**VecNorm**	**StopW**	**MinWFFreq**	**MinWFWeight**	**MinFFreq**			
	IslamInkpen 0.05	NoSmooth	L_2	false	NoLimit	Limit -0.2	NoLimit			
Cos-PPmi	**VecSim**		**Weight**			**FeatTransf**		**0.44**	**0.63**	**0.52**
	Cos		Pmi			NoTransf				
	DimRed	**Smooth**	**VecNorm**	**StopW**	**MinWFFreq**	**MinWFWeight**	**MinFFreq**			
	NoDimRed	NoSmooth	NN	false	NoLimit	Zero 0	NoLimit			
Cos-Pmi	**VecSim**		**Weight**			**FeatTransf**		**0.43**	**0.58**	**0.49**
	Cos		Pmi			NoTransf				
	DimRed	**Smooth**	**VecNorm**	**StopW**	**MinWFFreq**	**MinWFWeight**	**MinFFreq**			
	NoDimRed	NoSmooth	NN	false	NoLimit	NoLimit	NoLimit			

$$PearsMbAdjCosMod\text{-}3.Lb(u,v) = \begin{cases} 1, & d \geq 0.1 \\ \frac{d}{0.1}, & d < 0.1 \end{cases}$$

$$d = 0.5 \times \left(\frac{\sum_{i=1}^{n} sgn(u_i - \bar{u}) \times lb(|u_i - \bar{u}| + 1) \times sgn(v_i - \bar{v}) \times lb(|v_i - \bar{v}| + 1)}{lbinv\left(\sum_{i=1}^{n} (lb(|u_i - \bar{u}| + 1))^2\right)} + \right. \quad (5.1)$$
$$\left. \frac{\sum_{i=1}^{n} sgn(u_i - \bar{u}) \times lb(|u_i - \bar{u}| + 1) \times sgn(v_i - \bar{v}) \times lb(|v_i - \bar{v}| + 1)}{lbinv\left(\sum_{i=1}^{n} (lb(|v_i - \bar{v}| + 1))^2\right)} \right)$$

$$lbinv(x) = min(max(sgn(x) \times (2^{|x|} - 1), -2^{100}), 2^{100})$$

Further, the weighting scheme used in BestCvbmDcBnc2 is a combination of PMI_α (PmiAl) Levy et al. (2015), PMI with Laplace smoothing (PmiWls) Turney and Pantel (2010) and Unisubtuples (Unis) Pecina (2010):

$$PmiAl\text{-}Tc3Tw0S2P0(x,y) = lb\left(\frac{n'_\alpha \times f'_{xy}}{f'_x \times f'^{0.75}_y}\right) - 3.29 \times \sqrt{\frac{1}{a} + \frac{1}{b} + \frac{1}{c} + \frac{1}{d}}$$

$$a = f'_{xy}, \quad b = f'_x - f'_{xy}, \quad c = f'_y - f'_{xy}, \quad d = n' - f'_x - f'_y + f'_{xy}$$

$$f'_x = f_x + 1,\ f'_y = f_y + 1,\ f'_{xy} = f_{xy} + 1,\ n' = n + 1,\ n'_\alpha = \left(\sum_{i=1}^{|V|} f_i^{0.75}\right) + 1 \quad (5.2)$$

f_x, f_y: word frequencies, $f_{x,y}$: xy tuple frequency

n: total number of words in the corpus, $|V|$: size of the vocabulary

One should be able to have enough understanding of the settings of the other parameters in BestCvbmDcBnc2 from the information provided in Section 4.2.

5.2.2 Results using the semantic vectors of Mikolov et al. (2013b)

With the settings of the 4 parameters selected in the first phase, altogether 1632 configurations were tested. A small proportion of these together with their performance are presented in Table 5.2. The configuration with the best results is noted as BestPmMv2.

As already mentioned before, the number of possible settings of several parameters have been significantly increased since Dobó and Csirik (2019a) with numerous novel variants. The results of the best found predictive configuration with those reduced number of settings for several parameters will also be included in the rest of the thesis (BestPmMv).

Please note that although we were able to find a better configuration with our extended settings set for several parameters than the one presented in Dobó and Csirik (2019a), meaning that BestPmMv2 performs better that BestPmMv, the actual scores of these models presented here are actually lower than the best scores presented in Dobó and Csirik (2019a). This is due to a technical change in the evaluation process: previously we accepted our models to return non-zero similarity scores when at least one of the compared words had a zero lengths feature vector. Now we do not allow this, as we think that it seems theoretically more correct this way. However, some of the test words in the MD2 dataset are out of vocabulary in case of the Mv dataset as that dataset contains each word only with its American spelling, and the MD2 dataset contains words with both American and British spelling. Because of these out-of-vocabulary words the scores of every configuration are lower now, then as in Dobó and Csirik (2019a). This change only affects the ranking of the configurations in very rare cases, and only in a

Table 5.2. Performance of a selection of configurations from the heuristic analysis in the second phase using the Mv.

Abbrev	VecSim	FeatTransf	VecNorm	MinWFWeight	P	S	H
BestPmMv2	MbAdjCosLogProd	Weight BefNorm Lb	L_2	Limit -0.2	0.72	0.72	0.72
-	MbAdjCosLogProd	Weight AftNorm Lb	L_2	Limit -0.2	0.72	0.72	0.72
-	MbAdjCosLogProd	Weight BefNorm Lb	L_1	Limit -0.2	0.72	0.72	0.72
-	MbAdjCosLogProd	Weight AftNorm Sigm	L_1	Limit -0.2	0.71	0.72	0.72
-	MbAdjCosLogProd	NoTransf	L_1	Limit -0.2	0.71	0.72	0.72
-	MbAdjCosLogProd	Weight AftNorm Sigm	L_2	Limit -0.2	0.71	0.72	0.72
-	MbAdjCosLogProd	Weight BefNorm Sigm	L_2	Limit -0.2	0.71	0.72	0.72
-	MbAdjCosLogProd	Weight BefNorm Sigm	L_1	Limit -0.2	0.71	0.72	0.72
-	MbAdjCosLogProd	NoTransf	L_2	Limit -0.2	0.71	0.72	0.72
-	MbAdjCosLogProd	Weight AftNorm Lb	L_1	Limit -0.2	0.71	0.72	0.72
-	MbAdjCosLogProd	Weight BefNorm Lb	L_2	Zero -0.5	0.72	0.72	0.72
-	MbAdjCosLogProd	Weight BefNorm Lb	L_2	Limit -0.5	0.72	0.72	0.72
-	MbAdjCosLogProd	Weight BefNorm Lb	L_2	NoLimit	0.72	0.72	0.72
Cos	Cos	NoTransf	L_2	NoLimit	0.71	0.72	0.71
BestPmMv	SmoothCos	Weight AftNorm Sigm	L_2	NoLimit	0.68	0.72	0.70
Cos-Zero0	Cos	NoTransf	L_2	Zero 0	0.60	0.69	0.64
BestCvbmDcBnc2OnPm	PearsMbAdjCosMod-3.Lb	NoTransf	L_1	Zero 0	0.38	0.47	0.42

minor way, and it does not have any effect on which configuration was found best.

The results achieved with the selected best configuration in case of count-vector-based DSMs (BestCvbmDcBnc2; see Table 5.1; please note that only 4 out of the 10 parameters could be used here), run on information extracted from Mv, are also included in Table 5.2, and is noted as BestCvbmDcBnc2OnPm.

5.2.3 Verification of the heuristic approach

To verify our heuristic approach and its results, we have done a full analysis using the Mv with that of the heuristic analysis in Dobó and Csirik (2019a), and we have discovered that the best 26 configurations were the same using both analyses, which has highly exceeded our expectations. Unfortunately we were not able to repeat this full analysis now, as since then we have constructed many new settings for the tested parameters, which increased the number of possible configurations to a level that made performing a full analysis unmanageable. However,

we think that if we were able to do it, then we would come to the same conclusions as in Dobó and Csirik (2019a).

Further, later on we have done numerous tests with our found best configurations in such a way that they were tested using several different input data types, both in case of CVBM and PM configurations. The result of a number of these tests will later be displayed in 5.3, where one can see that all CVBM and PM configurations performed well when using any input data that was of the same type as the original input data used to create the configuration.

Both these results give us a very strong verification that the idea behind our heuristic approach was good, and that its results are reliable. Moreover, it also supports our assumption that a configuration working well on given input data also works well on other input data of the same type.

5.3 Evaluation and discussion of results for English

In this section we evaluate the results of our best configurations for count-vector-based, predictive and knowledge-graph-based models, determined using our heuristic approach, with the help of multiple input data sources and multiple test datasets.

5.3.1 Evaluation of our best configurations on the MD2 development dataset

Our results on the MD2 dataset (see Tables 5.1 and 5.2) show that the BestCvbmDcBnc2 and the BestPmMv2 configurations, both incorporating novel parame-

ter settings, significantly outperform conventional variants (e.g. simple methods with cosine similarity and positive pointwise mutual information).

Although the parameter settings of the BestCvbmDcBnc2 and BestPmMv2 configurations resemble each other to some extent, their 4 mutual parameters do not have the exact same settings. Further, the two mixed configurations (BestCvbmDcBnc2OnPm and BestPmMv2OnCvbm) perform significantly worse than the BestCvbmDcBnc2 and BestPmMv2. Out of the BestCvbmDcBnc2 and BestPmMv2 variants the latter achieved slightly better results.

Although all of our best configurations were selected based on a heuristic approach, in Section 5.2.3 we were able to verify that our idea behind this approach is good and that its results are sound and reliable. Further, we have also validated our assumption that the same parameters configuration can be successfully used in case different data sources of the same type are used as input.

5.3.2 Evaluation of our best configurations on the MT dataset, using multiple sources as input

All 4 of our previously inspected configurations, as well as the best configurations found in Dobó and Csirik (2019a), have also been tested on the MT test dataset (see Table 5.3). We have evaluated the same configurations using multiple input data sources (i.e. different extracted raw counts or different semantic vectors) in each case.

Similarly as in case of the result on the MD2 dataset (second phase), the results of the BestPmMv2UsingMv model are slightly superior to that of the BestCvbmDcBnc2UsingDcBnc model. Further, as anticipated, the mixed models using

Table 5.3. Performance of our best models on the MT dataset. The methods are grouped into 3 categories based on the type of input data used.

Input data	Configuration	P	S	H
CVBMs using solely distributional linguistic data				
DcBnc	BestPmMv2OnCvbm	0.51	0.64	0.57
	BestCvbmDcBnc	0.67	0.69	0.68
	BestCvbmDcBnc2	0.71	0.71	0.71
DcEw	BestCvbmDcBncMF20	0.70	0.73	0.71
	BestCvbmDcBnc2MF20	0.74	0.73	0.73
DcUkwac	BestCvbmDcBncMF100	0.76	0.77	0.76
	BestCvbmDcBnc2MF100	0.74	0.75	0.75
LcBnc	BestCvbmDcBnc	0.65	0.69	0.67
	BestCvbmDcBnc2	0.61	0.61	0.61
LcEw	BestCvbmDcBncMF20	0.71	0.75	0.73
	BestCvbmDcBnc2MF20	0.71	0.71	0.71
LcUkwac	BestCvbmDcBncMF100	0.75	0.77	0.76
	BestCvbmDcBnc2MF100	0.75	0.74	0.74
EcBnc	BestCvbmDcBnc	0.72	0.74	0.73
	BestCvbmDcBnc2	0.67	0.67	0.67
EcEw	BestCvbmDcBncMF20	0.74	0.78	0.76
	BestCvbmDcBnc2MF20	0.72	0.72	0.72
EcUkwac	BestCvbmDcBncMF100	0.78	0.79	0.78
	BestCvbmDcBnc2MF100	0.73	0.74	0.73
PMs using solely distributional linguistic data				
Mv	BestCvbmDcBnc2OnPm	0.33	0.40	0.36
	BestPmMv	0.70	0.73	0.71
	BestPmMv2	0.73	0.73	0.73
Bv	BestPmMv	0.78	0.80	0.79
	BestPmMv2	0.79	0.79	0.79
Other types of models				
Sv	BestPmMv	0.85	0.87	0.86
	BestPmMv2	0.85	0.85	0.85
	BestSv	0.85	0.87	0.86
	BestSv2	0.87	0.87	0.87

both the DcBnc and the Mv as input (BestPmMv2OnCvbmUsingDcBnc and BestCvbmDcBnc2OnPmUsingMv) achieved worse results than the non-mixed model using the same input data (BestCvbmDcBnc2UsingDcBnc and BestPmMv2UsingMv,

respectively) in all cases. On the other hand all count-vector-based and predictive configurations performed well on all input data of the same type, as already discussed in Section 5.2.3.

As the vectors in Sv are not predictive vectors, as opposed to the other used semantic vectors, but are rather constructed from a knowledge graph, it was anticipated that they might behave differently than the other vectors. Therefore, in the end, we have decided to run the same two-phase heuristic analysis using the Sv as we have done using the Mv. As a result of this, we have got the following parameter settings to perform best on these vectors, using the MD2 dataset (BestSv2):

- the LMod-9.1.Cu similarity measure
- P1D2 feature transformation of the weights before normalization
- L_2 vector normalization
- the "Limit -0.2" option on word-feature tuple weights

The results of the best found configuration using Sv as input with the reduced number of settings for several parameters tested in Dobó and Csirik (2019a) will also be included in the rest of the thesis (BestSv). The BestSv2 configuration achieved the highest scores on the MT dataset, using the Sv as underlying data. The H score of this model is also bit higher than that of the BestPmMv2UsingSv model, which was expected as in case of the former the parameters were optimized on the same input that was used as underlying data source for the model.

When comparing our new best CVBM configuration (BestCvbmDcBnc2) to that presented in Dobó and Csirik (2019a) (BestCvbmDcBnc), we can usually see

an improvement when using the Dc information extraction method, but a decline in case of using the Lc and Ec extraction methods. This suggests us that the BestCvbmDcBnc configuration is a more general one that performs well for any type of CVBM input data. Our new one still performs decently in case of any CVBM input, however, it seems to be somewhat specialized for the Dc information extraction method. Based on this both our old and new configurations can be useful in the future, for different types of input, and it is worth continuing the experiments with both. In case of the PM models, the new configuration (BestPmMv2) seems to be slightly better than the previous version (BestPmMv). When looking at the results of these configurations using the Sv as underlying data, the opposite seems to be true.

When using the Ew and the ukWaC as underlying corpus with the BestCvbmDcBnc and BestCvbmDcBnc2 configurations, we had to limit the words and features to those having a minimum frequency of 20 and 100 (noted as MF20 and MF100 in the model names), respectively, due to computational reasons. Based on the result presented in Section 5.1.1.10 and also verified by additional tests, such limitations have a negative impact on the results. Despite this negative effect, it is clear that the larger the used corpus is the better the results are, having the information extracted with either the Dc, Lc or Ec method. We were not able to do the same comparison in case of PMs as input, as all models were produced using different input data, but most likely we would have come to the same conclusion as in case of CVBMs.

In Dobó and Csirik (2019a) we have noted that when using raw counts as input with the same underlying corpus in all cases, then the models using the information extraction method of Salle et al. (2016a) produce better results than the

ones using the method of Levy et al. (2015), which in turn mostly produce better results than the ones using the method of Dobó and Csirik (2013). As beside the used information extraction method all other properties of the compared models are the same, this implies that the data extraction method of Salle et al. (2016a) is superior to that of Levy et al. (2015), which is in turn superior to that of Dobó and Csirik (2013). This does not seem to hold with respect to the new BestCvbmDcBnc2 configuration, but the BestCvbmDcBnc2 seems to be a bit specialized to the Dc information extraction method, so it might mostly be due to that.

In case of semantic vectors as input, using the Sv semantic vectors as input produces the best results, followed by the models using Bv and Mv, in that order, which is in line with the original results presented for those models (see Table 5.4).

As our best models achieved very similar results on all development and test datasets as well as using different input data, we can conclude that there was no overfitting.

5.3.3 Comparison of our best results with the state-of-the-art

Beside the MT dataset, we have also run our best models on several other test datasets, and compared our results to that of state-of-the-art models in Table 5.4. In our view the best evaluation metric is the H score, as it takes both the similarity scores and the rankings returned by the models into account. Unfortunately, in case of most methods from other studies, only the P or the S value was reported. In such cases it was not possible to determine the H score for the method, and thus it was only possible to compare these results with ours using the reported

scores, and not the H score.

Table 5.4. Performance of our best models and some state-of-the-art systems on the test datasets, evaluated on the test datasets with the help of the Pearson (P) and Spearman (S) correlation coefficients, as well as the H scores calculated from them. Please note that the results on the RG, MC, WC and TO datasets are rather unreliable, so conclusions based on them should be taken cautiously, as also noted in Section 3.2. The results for the models marked with * come from reproductions of the given model by us, to be able to report all scores for those models. (In case of the model of Yin and Schütze (2016) this was also necessary as the results reported in the original article were produced using only those words that were in the vocabulary of their model, and not on the full test datasets.)

Test dataset	**MT**			**MF**			**RG**			**MC**			**WS**			**SL**			**TO**
Evaluation measure	**P**	**S**	**H**	**P**	**S**	**H**	**P**	**S**	**H**	**P**	**S**	**H**	**P**	**S**	**H**	**P**	**S**	**H**	**A**
Count-vector-based models using solely distributional linguistic data																			
Kiela and Clark (2014)	-	-	-	-	0.71	-	-	0.74	-	-	0.65	-	-	0.58	-	-	-	-	0.83
Baroni et al. (2014)	-	0.72	-	-	-	-	0.70	-	-	-	-	-	-	0.62	-	-	-	-	0.76
De Deyne et al. (2017)	-	-	-	-	0.75	-	-	0.78	-	-	-	-	-	-	-	-	0.37	-	-
Iosif et al. (2016)	-	0.76	-	-	0.76	-	-	-	-	-	-	-	-	0.70	-	-	-	-	-
Salle et al. (2016a)	-	-	-	-	0.76	-	-	0.79	-	-	0.82	-	-	-	-	-	0.34	-	-
Levy et al. (2015)	-	-	-	-	0.78	-	-	-	-	-	-	-	-	-	-	-	0.43	-	-
Pennington et al. (2014)*	0.80	0.80	0.80	0.80	0.80	0.80	0.77	0.77	0.77	0.81	0.83	0.82	0.70	0.71	0.71	0.43	0.41	0.42	0.90
Salle et al. (2018)*	0.81	0.81	0.81	0.80	0.81	0.81	0.79	0.76	0.78	0.82	0.82	0.82	0.70	0.73	0.72	0.43	0.42	0.42	0.80
BestCvbmDcBncUsingDcBnc	0.67	0.69	0.68	0.67	0.70	0.69	0.80	0.81	0.81	0.80	0.81	0.81	0.56	0.57	0.56	0.37	0.37	0.37	0.78
BestCvbmDcBnc2UsingDcBnc	0.71	0.71	0.71	0.71	0.70	0.70	0.74	0.73	0.73	0.83	0.85	0.84	0.58	0.58	0.58	0.40	0.39	0.40	0.81
BestCvbmDcBncMF100UsingEcUkwac	0.78	0.79	0.78	0.77	0.78	0.78	0.77	0.78	0.77	0.69	0.69	0.69	0.54	0.56	0.55	0.34	0.34	0.34	0.81
BestCvbmDcBnc2MF100UsingEcUkwac	0.73	0.74	0.73	0.72	0.74	0.73	0.60	0.65	0.62	0.56	0.58	0.57	0.58	0.58	0.58	0.36	0.34	0.35	0.70
Predictive models using solely distributional linguistic data																			
Wieting et al. (2016)	-	-	-	-	-	-	-	-	-	-	-	-	-	0.58	-	-	0.71	-	-
Hill et al. (2014a)	-	0.63	-	-	-	-	-	-	-	-	-	-	-	0.57	-	-	0.52	-	0.93
Yin and Schütze (2016)*	0.74	0.73	0.73	0.73	0.72	0.72	0.79	0.79	0.79	0.86	0.88	0.87	0.65	0.67	0.66	0.48	0.47	0.47	0.88
Iosif et al. (2016)	-	0.74	-	-	0.75	-	-	-	-	-	-	-	-	0.68	-	-	-	-	-
De Deyne et al. (2017)	-	-	-	-	0.79	-	-	0.83	-	-	-	-	-	-	-	-	0.43	-	-
Baroni et al. (2014)*	0.78	0.80	0.79	0.78	0.80	0.79	0.83	0.84	0.84	0.84	0.84	0.84	0.68	0.73	0.71	0.46	0.46	0.46	0.89
Christopoulou et al. (2018)	-	0.84	-	-	-	-	-	-	-	-	-	-	-	0.73	-	-	-	-	-
BestPmMvUsingMv	0.70	0.73	0.71	0.70	0.73	0.71	0.77	0.76	0.77	0.81	0.82	0.81	0.63	0.68	0.65	0.45	0.44	0.44	0.88
BestPmMv2UsingMv	0.73	0.73	0.73	0.73	0.73	0.73	0.77	0.76	0.76	0.81	0.81	0.81	0.65	0.68	0.67	0.46	0.44	0.45	0.88
BestPmMvUsingBv	0.78	0.80	0.79	0.78	0.80	0.79	0.83	0.84	0.84	0.84	0.84	0.84	0.68	0.73	0.71	0.46	0.46	0.46	0.90
BestPmMv2UsingBv	0.79	0.79	0.79	0.78	0.79	0.79	0.84	0.84	0.84	0.84	0.82	0.83	0.70	0.73	0.71	0.47	0.46	0.46	0.90
Other types of models																			
Faruqui and Dyer (2015)	-	-	-	-	-	-	-	0.67	-	-	-	-	-	0.45	-	-	0.58	-	-
Banjade et al. (2015)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.65	0.64	0.65	-
Mrkšić et al. (2017)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.75	-	-
Recski et al. (2016)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.76	-	-
Vulić et al. (2017)	-	-	-	-	-	-	-	-	-	-	-	-	-	0.76	-	-	0.78	-	-
Yih and Arbor (2012)	-	-	-	-	-	-	-	0.89	-	-	-	-	-	0.81	-	-	-	-	-
Lazaridou et al. (2015)	-	0.75	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.37	-	-
Rothe and Schütze (2017)	-	-	-	-	0.78	-	-	0.83	-	-	0.85	-	-	0.69	-	-	0.47	-	-
Bruni et al. (2013)	-	0.78	-	-	-	-	-	-	-	-	-	-	-	0.72	-	-	-	-	-
Collell et al. (2017)	-	-	-	-	0.81	-	-	-	-	-	-	-	-	0.69	-	-	0.41	-	-
Lee et al. (2016)	-	-	-	-	0.83	-	-	0.92	-	-	-	-	-	0.79	-	-	-	-	-
De Deyne et al. (2017)	-	-	-	-	0.87	-	-	0.95	-	-	-	-	-	-	-	-	0.68	-	-
Speer et al. (2017)*	0.85	0.87	0.86	0.85	0.86	0.85	0.90	0.90	0.90	0.87	0.89	0.88	0.76	0.82	0.79	0.63	0.62	0.62	0.99
BestPmMvUsingSv	0.85	0.87	0.86	0.85	0.86	0.85	0.90	0.90	0.90	0.87	0.89	0.88	0.76	0.82	0.79	0.63	0.62	0.62	0.99
BestPmMv2UsingSv	0.85	0.85	0.85	0.85	0.85	0.85	0.89	0.86	0.88	0.86	0.86	0.86	0.77	0.82	0.80	0.64	0.62	0.63	0.99
BestSvUsingSv	0.85	0.87	0.86	0.85	0.86	0.85	0.89	0.90	0.90	0.87	0.89	0.88	0.35	0.82	0.49	0.53	0.62	0.57	0.99
BestSv2UsingSv	0.87	0.87	0.87	0.86	0.86	0.86	0.91	0.90	0.91	0.89	0.89	0.89	0.28	0.80	0.41	0.48	0.62	0.54	0.99

As expected from the results presented in Table 5.3, the scores of the BestSv2 configuration using the Sv as input are the highest in case of most test datasets. However, it was interesting to see the significant drop in the P and thus the H score of the former two when tested on the WS dataset, for which we do not

yet have a good explanation. Further, this configuration has mixed results when tested using predictive semantic vectors as input, which further shows that these vectors are different from predictive vectors. On the other hand, our best predictive configuration (BestPmMv2) performs rather well using the Sv as input vectors too, although mostly slightly underperforming BestSv2, which was expected as in case of the BestSv2 the parameters were optimized on the same input that was used as underlying data source for the model.

When comparing the results of our new and previous (Dobó and Csirik, 2019a) best CVBM and PM models, the same conclusions mostly seem to hold in case of all test datasets, as we have concluded from Table 5.3 for the MT dataset.

While all other datasets are based on relatedness, the SL dataset contains similarity scores, as also noted in Section 3.1. Most of our models achieve much lower scores on this dataset than on the other test datasets. However, all our configurations using count-vector-based or predictive input data have a rather low performance on this dataset, while the same configurations using the knowledge-graph-based Sv as input usually achieve considerably better results. So the usually lower results on the SL dataset seems to mostly depend on the input data used, and not the chosen configuration.

Our best model overall (BestSv2UsingSv) achieved better results than any previous model on the most important test datasets. On the other hand, when considering only PMs, our best results are a little lower than that of Christopoulou et al. (2018). However, as we were unable to acquire the model of Christopoulou et al. (2018), we had to use the semantic vectors of the second best PM (at least according to our knowledge) as input, namely that of Baroni et al. (2014). We were able to achieve basically the same results with our best predictive configu-

ration on the Bv vectors as Baroni et al. (2014) with their original configurations. However, our results are not directly comparable with that of Christopoulou et al. (2018) due to the differences in the used input data.

Further, when only looking at CVBMs, our best results are also a little lower than previous state-of-the-art. However, the underlying corpora used in the previous state-of-the-art model is approximately 30 times as large as the one used in our best models. Unfortunately, it would have been unmanageable to test our configurations (not to mention running our whole heuristic analysis) with such large corpora with the resources available to us. As the size of the used corpus has a large impact on the results, as it could also be seen in Table 5.3, our best configurations could only be reliably compared to that of others if the used input data was also the same, which is not true here.

So, it is not possible to reliably compare our best configurations with previous state-of-the-art configurations based on the results reported in Table 5.4 in case of PMs and CVBMs. Thus, these can only be viewed as supplementary results, while conclusions should not be taken from them. Therefore, in order to be able to present a reliable comparison, we have done tests with multiple state-of-the-art configurations in such a way that the same input data was used for both those configurations and for our ones too (see Table 5.5). For this, we have used state-of-the-art count-vector-based methods to extract information from different corpora, and ran tests using these as input, as well as testing on some state-of-the-art semantic vectors.

In those cases where count-vector-based extracted information was used as input (Lc* and Ec*), all 10 of our inspected parameters could be tested. Where directly the obtained semantic vectors were used as input (Ev, Bv, Pv, Sv), only

4 of the 10 considered parameters could be tested as the other parameters could have only been used during the construction of the vectors, which was already done. In case of CVBM semantic vectors (Pv and Ev), this unfortunately reduced the number of parameters that could be tested from 10 to 4. In case of the other semantic vectors it did not result in any disadvantages, as in case of such models we could only have used the same 4 parameters by default anyway.

In case of semantic vectors used as input, our best results were always at least as good as that of the original configuration (OSC) proposed by the authors for the given model, with a slight advantage in a couple of cases. On the other hand, when using the count-vector-based information extracted from different corpora, we achieved better results than the OSC proposed both by Levy et al. (2015) and by Salle et al. (2016a) in their state-of-the-art model in case of all input data, with a considerable margin in most cases. During these tests too, using larger input corpora clearly improved the results.

5.3.4 Discussion of results for English

During our heuristic approach we were able to find such novel configurations using the counts of Dobó and Csirik (2013) on the British National Corpus, the predictive semantic vectors of Mikolov et al. (2013b) and the semantic vectors of Speer et al. (2017) constructed from a knowledge graph, incorporating novel parameter settings in all three, that significantly outperform conventional configurations.

Out of the best models found for the different input data, the one using the semantic vectors of Speer et al. (2017) achieves considerably better results than

Table 5.5. Comparison of our best configurations with state-of-the-art models, with the original configuration (OSC) proposed by the authors for those models, using the same input data for the OSCs and for our best configurations, evaluated on the MT dataset.

Input data	Configuration	# of tested parameters	P	S	H
CVBMs using solely distributional linguistic data					
LcBnc	OSC	-	0.61	0.64	0.63
	Cos-PPmi	-	0.52	0.57	0.55
	BestCvbmDcBnc	10	0.65	0.79	0.67
	BestCvbmDcBnc2	10	0.61	0.61	0.61
LcEw	OSC	-	0.69	0.73	0.71
	Cos-PPmi	-	0.60	0.69	0.64
	BestCvbmDcBncMF20	10	0.71	0.75	0.73
	BestCvbmDcBnc2MF20	10	0.71	0.71	0.71
LcUkwac	OSC	-	0.72	0.73	0.73
	Cos-PPmi	-	0.62	0.70	0.66
	BestCvbmDcBncMF100	10	0.75	0.77	0.76
	BestCvbmDcBnc2MF100	10	0.75	0.74	0.74
EcBnc	OSC	-	0.59	0.58	0.58
	Cos-PPmi	-	0.56	0.58	0.57
	BestCvbmDcBnc	10	0.72	0.74	0.73
	BestCvbmDcBnc2	10	0.67	0.67	0.67
EcEw	OSC	-	0.62	0.62	0.62
	Cos-PPmi	-	0.60	0.66	0.63
	BestCvbmDcBncMF20	10	0.74	0.78	0.76
	BestCvbmDcBnc2MF20	10	0.72	0.72	0.72
EcUkwac	OSC	-	0.76	0.77	0.77
	Cos-PPmi	-	0.61	0.63	0.62
	BestCvbmDcBncMF100	10	0.78	0.79	0.78
	BestCvbmDcBnc2MF100	10	0.73	0.74	0.73
Pv	OSC (Cos)	-	0.80	0.80	0.80
	Cos-Zero0	-	0.73	0.74	0.74
	BestPmMv	4	0.80	0.80	0.80
	BestPmMv2	4	0.82	0.82	0.82
Ev	OSC (Cos)	-	0.81	0.81	0.81
	Cos-Zero0	-	0.77	0.78	0.78
	BestPmMv	4	0.81	0.81	0.81
	BestPmMv2	4	0.79	0.79	0.79
PMs using solely distributional linguistic data					
Bv	OSC (Cos)	-	0.78	0.80	0.79
	Cos-Zero0	-	0.74	0.76	0.75
	BestPmMv	4	0.78	0.80	0.79
	BestPmMv2	4	0.79	0.79	0.79
Other types of models					
Sv	OSC (Cos)	-	0.85	0.87	0.86
	Cos-Zero0	-	0.82	0.83	0.82
	BestPmMv	4	0.85	0.87	0.86
	BestPmMv2	4	0.85	0.85	0.85
	BestSv	4	0.85	0.87	0.86
	BestSv2	4	0.87	0.87	0.87

the other two, and our best predictive model has only slightly higher scores than our best count-vector-based model. It is clear from our results that different parameter settings have to be used in case of different types of input. When looking at our count-vector-based model compared to the other two models, then most likely this is partially due to the fact that the settings of the different parameters influence each other, and in case of using semantic vectors as input only 4 of the 10 parameters could be used. On the other hand, a configuration working well using a given count-vector-based or predictive input also works well using other input of the same type.

It is clear from the results that most of our models are much less suitable to determine the similarity of two words, than they are to determine their relatedness. However, we could conclude from the experiments that this mostly depends on the input data, and not the used configuration, meaning that with input data tailored for similarity our best configurations would be most likely successful in determining the similarity of words.

It is easy to see that using a larger corpus as input for count-vector-based models produces considerably better results, despite having to reduce the number of words and features used due to computational reasons. This is most likely true for predictive models too.

Our best model overall, having the BestSv2 configuration and using the Sv vectors constructed from a knowledge graph as input, achieved state-of-the-art results surpassing all previous models on the most important test datasets. On the other hand, when considering only CVBMs and PMs, our best results are a little lower than previous state-of-the-art. However, it would have been unmanageable to test our configurations (not to mention running our whole heuristic

analysis) with such large corpora as used in the previous state-of-the-art CVBM model with the resources available to us. Further, we were not able to acquire the model of Christopoulou et al. (2018) to try their semantic vectors as input.

Unfortunately, different configurations can only be reliably compared to each other if the used input data is the same for all of them. Therefore, our most important tests were those where we used the same input data for state-of-the-art configurations and for our newly proposed configurations too. During these tests, with any set of semantic vectors as input, our best results were always at least as good as that of the state-of-the-art original configuration proposed by the authors of the given model, with a slight advantage in a couple of cases, even though in these cases only 4 of our 10 examined parameters could be tested.

Moreover, with our novel combination of the settings of the 10 parameters tested when extracted information of state-of-the-art count-vector-based models were used, we could clearly outperform the original configurations in case of all input data, with a considerable margin in most cases. These reflect our previous results and intuition, as experienced during our heuristic analysis.

Based on these results we believe that our best CVBM and PM configurations could also achieve absolute state-of-the-art results in their category if they were used with the same input data as previous state-of-the-art models. Unfortunately, testing these was not possible within this research.

The best configurations found in the second phase of our heuristic approach are not simply the combinations of the best parameter settings found during the first phase, when the parameters were tested one by one. Moreover, by including further possible settings for several parameters in our analysis in Dobó and Csirik (2019b) compared to Dobó and Csirik (2019a), several parameter settings in our

newly found best configuration are considerably different compared to the best configuration found in Dobó and Csirik (2019a). These clearly show that our intuition was correct that the settings of the different parameters are dependent on each other, and instead of testing the parameters separately they need to be tested together, also considering the interactions between them.

To sum up, the main findings of our analysis are that

- we could outperform previous state-of-the-art results when using raw counts as input and thus all 10 parameters could be optimized,
- we were able to find such configurations that perform at least as well, with a slight superiority in a couple of cases, as previous state-of-the-art models, when using semantic vectors as input and thus only 4 out of 10 parameters could be optimized,
- our best model, BestSv2UsingSv, based on semantic vectors constructed from a knowledge graph, achieves absolute state-of-the-art results compared to all previous models of any type on the most important test datasets.

CHAPTER 6

Comparison of our findings for English, Spanish and Hungarian

To be able to compare our findings for the different languages, we have done the same extensive analysis for Spanish and Hungarian as for English (described in Section 4.1 in detail, and compared the findings of these. For reproducibility and transparency, we plan to make our most important data, code and results with respect to all languages publicly available at:

https://github.com/doboandras/dsm-parameter-analysis/.

6.1 Results of the first phase

As also described in Section 5.1, during the first phase of our analysis multiple runs were done for each setting of every parameter, and the most promising ones in case of each parameter were selected to be included in the second phase. In case of English, we used half of the development part of the MEN dataset for evaluation, while for Spanish the Spanish WordSimilarity-353 dataset and for Hungarian half of the Hungarian TOEFL dataset was employed. The top 5 performing settings for each parameter are listed in Table 6.1 in case of each language.

6.2 Results of the second phase

In the second phase all possible combinations of the selected settings of each parameter were tested in case of all three languages, in order to find the best configuration for all languages, as also described in Section 5.2. This meant testing 40860, 44544 and 28576 configurations for English, Spanish and Hungarian, respectively. The second half of the development part of the MEN dataset was used for testing in case of English, while the Moldovan dataset and the second part of the Hungarian TOEFL dataset were used for Spanish and Hungarian, respectively. A selection of the second phase results for the three languages are presented in Tables 5.1, 6.2 and 6.3, respectively.

6.3 Results on the test datasets

The best configuration for English was tested on the test part of the MEN dataset (MT), and the best configurations for Spanish and Hungarian were tested on the respective version of the Rubenstein-Goodenough dataset (RG) to give us the final results. The best configuration of each language was also evaluated on the datasets of the other languages, to provide us a way of comparison. The results of these test can be found in Table 6.4.

6.4 Evaluation and discussion

In this section we evaluate our results presented in the previous sections. Please note that the scores are not fully comparable across languages, even when considering the same datasets on different languages, as except for the Moldovan dataset all of the used Spanish and Hungarian datasets were constructed by translating the English versions, and thus the results on them can be distorted and less reliable than on their English counterparts. Furthermore, the Spanish and Hungarian datasets, especially the latter ones, are rather small, which also makes them less reliable than the English ones.

As there are many differences in the syntax and morphology of the different languages, we anticipated from the beginning that there will be at least some small differences in our findings for the different languages. However, our intuition was that our findings for the different languages will be subtle, and we will be able to find good and rather language-independent configurations. As English and Spanish belong to the family of Indo-European languages, while Hungarian

does not, we expected that the results for English and Spanish will be similar due to this. Further, as both Spanish and Hungarian have very rich morphology, we expected that there will also be a higher similarity between our results for Spanish and Hungarian because of this. We anticipated that the least similarities will be between English and Hungarian, as these languages are the least similar to each other.

In the first phase of our analysis we could observe that some of the parameters worked exactly the same way or very similarly across languages. These parameters were the weighting scheme, feature transformation, vector normalization and minimum limits on word-feature frequencies. These findings are in line with our initial intuitions. Dimensionality reduction seemed to be similar for English and Spanish, while a bit different for Hungarian. Smoothing seemed to perform similarly for Spanish and Hungarian, while differently for English. Minimum limits on word-feature weights seem to behave a bit differently for all three languages. However, it was interesting to see that the results for vector similarity measures, stop-word filtering and minimum limits on feature frequencies were rather similar for English and Hungarian, but different for Spanish, which is contrary to what we anticipated.

In the second phase, although there were similarities in the found best configurations across the different languages, one could also observe many differences. Here too, the weighting schemes, feature transformation and minimum limits on word-feature frequencies were mostly similar. Compared to the first phase vector similarity, smoothing and minimum limits on feature frequencies were also alike for all languages. The other parameters showed a different behavior for at least one language compared to the others.

As also noted in Section 5.2.1, there were actually two distinct configurations with the same best score for English, and they were only different in their DimRed parameter setting. We have chosen the one with the "IslamInkpen 0.05" setting as best English configuration, as that setting achieved better performance in the first phase than the "NoDimRed" setting in the other configuration. Furthermore, for Hungarian there were four configurations with the same best score. We have used a similar approach in selecting the best version, as we have done in case of the English version. However, as these different configurations with the same best results have different settings in case of some parameters, one has to be careful drawing conclusions from the best configurations of the different languages, and thus any conclusions drawn from them should be taken with some reservations.

The final conclusions for the parameters are the following:

- VecSim: for all languages measures based on cosine similarity achieve the best results
- Weight: measures based on PMI dominate the top of the table by far in case of all languages
- FeatTransf: no transformation and transforming the word-feature weights after normalization preforms best for all languages
- DimRed: dimensionality reduction seems to help in most situations: while in case of English the IslamInkpen version performed the best alongside no dimensionality reduction, for Spanish and Hungarian SVD is superior to these options

- Smooth: the no smoothing option clearly outperforms all others for all languages
- VecNorm: for English the L_1 option clearly seems to be the best, while for Spanish and Hungarian the best configurations use either L_1 or L_2 normalization, and most configurations achieve the same or very similar results with either
- StopW: stop-word filtering seems to improve the results to some extent in case of Spanish, while it does not in case of English and Hungarian
- MinWFFreq: no limit is by far superior to the other options for all languages
- MinWFWeight: no limit seems to be the best option in case of Spanish and Hungarian, while the Zero option with different parameters seems to excel in case of English
- MinFFreq: a low limit or no limit seems to be best in case of all languages (as noted before, in case of SVD for Spanish we had to use a limit of 3 instead of no limit for computational reasons)

As we anticipated, there were parameters where the results for Spanish and Hungarian were similar, but different for English. However, it was interesting that we did not find any parameters that were alike for English and Spanish, but different for Hungarian. Further, to our surprise we found such a parameter, where the results were similar for English and Hungarian, but different for Spanish. These latter findings were in contrast to our initial intuition.

Although all Spanish scores in the second phase are much lower than the English and Hungarian ones, these are almost completely due to the dataset used,

and do not mean that the found Spanish configurations are worse than their English and Hungarian counterparts, as it was noted in the beginning of this section and can be seen from our results on the test datasets (see Table 6.4) too. It simply suggests that the dataset used for Spanish in this phase is considerably tougher than the ones used for English and Hungarian.

It was interesting to see that in the cross-language experiments on the test datasets the order of the best configurations of the different languages with respect to their performance is different in case of the datasets of the three languages. The best English configuration was always superior to its Spanish counterpart, but it has no absolute superiority over the best Hungarian configuration. Further, there is also no clear ranking between the best Spanish and Hungarian configurations. It was also interesting to see that in case of the Spanish dataset, although the best Spanish configuration achieved rather good results, actually it achieved the lowest score out of the three best configurations tested. It was the same for the best Hungarian configuration on the Hungarian dataset too.

All in all, there seems to be no clear ranking between the best configurations of the different languages, and all of them achieved good results on the datasets of all languages. So, although we got different best configurations for the different languages, all of them seem to be rather language-independent. These findings give us a strong intuition that our heuristic approach was good, and that our found best configurations for all languages and their results are robust and reliable.

The best configurations found in the second phase are not simply made up of the best parameter settings in the first phase in case of Spanish and Hungarian either. This further proves that our intuition was correct, and the parameters of

DSMs need to be tested simultaneously, rather than separately.

Table 6.1. The top 5 performing setting for each parameter in case of all 3 languages, in descending order of H scores

Parameter	English		Spanish		Hungarian	
	Setting	**H**	**Setting**	**H**	**Setting**	**H**
VecSim	PearsMbAdjCosMod-3.Lb	0.71	LinHindleRMod-7.1.2.Cu	0.37	PearsMbMod-1.Lb	0.80
	PearsMbAdjCosMod-4.Lb	0.71	LinHindleRMod-6.1.2.Cu	0.36	PearsMbMod-4.Lb	0.80
	PearsMbAdjCosMod-2.Lb	0.71	LinHindleRMod-1.1.2.Cu	0.36	MbMod-6.Lb	0.80
	PearsMbAdjCosMod-6.Lb	0.71	LinHindleRMod-7.1.2.Sq	0.36	PearsMbMod-5.Lb	0.80
	PearsMbAdjCosMod-6.Sigm	0.71	LinHindleRMod-3.1.2.Sq	0.36	PearsMbMod-2.Lb	0.80
Weight	PmiAl-Tc3Tw0S2P4	0.70	PmiAlUnis-Tc4Tw3S2P2	0.38	PmiAl-Tc4Tw2S1P1	0.85
	PmiAl-Tc3Tw0S2P0	0.70	PmiAlUnis-Tc4Tw3S2P1	0.38	PmiAlUnisAm-Tc0Tw3S2P1	0.85
	PmiAlUnis-Tc3Tw0S0P4	0.70	PmiAlUnis-Tc4Tw2S1P1	0.38	PmiAlUnisAm-Tc0Tw2S2P2	0.85
	PmiAlUnis-Tc3Tw0S0P0	0.70	PmiAlUnisAm-Tc4Tw3S2P5	0.38	PmiAl-Tc4Tw3S0P2	0.85
	PmiAl-Tc4Tw0S2P5	0.70	PmiAlUnis-Tc4Tw2S1P2	0.38	PmiAlUnisAm-Tc0Tw2S2P1	0.85
FeatTransf	Weight AftNorm Lb	0.67	Weight AftNorm Lb	0.34	Weight AftNorm Sqrt	0.85
	Freq Sq	0.67	Weight AftNorm Sigm	0.34	Weight BefNorm Sqrt	0.85
	Weight BefNorm Sigm	0.67	NoTransf	0.34	Weight AftNorm Sigm	0.80
	Weight AftNorm Sigm	0.67	Weight BefNorm Lb	0.34	NoTransf	0.80
	NoTransf	0.67	Weight BefNorm Sigm	0.34	Weight AftNorm Lb	0.80
DimRed	SVD 200	0.70	SVD 100	0.37	IslamInkpen 0.025	0.80
	SVD 100	0.70	SVD 200	0.36	IslamInkpen 0.25	0.80
	SVD 300	0.69	SVD 500	0.35	SVD 200	0.80
	SVD 500	0.68	SVD 300	0.34	IslamInkpen 0.005	0.78
	IslamInkpen 0.05	0.67	IslamInkpen 0.01	0.34	IslamInkpen 0.01	0.78
Smooth	NoSmooth	0.67	Freq KNS	0.34	Freq KNS	0.83
	Weight KNS	0.65	Freq MDKNSPOMD	0.34	Freq MDKNSPOMD	0.80
	Freq KNS	0.62	NoSmooth	0.34	NoSmooth	0.78
	Freq MDKNSPOMD	0.62	Freq MKNS	0.33	Weight KNS	0.75
	Freq MKNS	0.57	Weight KNS	0.31	Freq MKNS	0.73
VecNorm	L_2	0.67	L_2	0.34	L_2	0.80
	L_1	0.67	L_1	0.34	NN	0.78
	NN	0.67	NN	0.34	L_1	0.75
StopW	false	0.67	true	0.34	false	0.80
	true	0.67	false	0.34	true	0.75
MinWFFreq	NoLimit	0.67	NoLimit	0.34	NoLimit	0.80
	2	0.60	2	0.30	3	0.68
	3	0.57	3	0.29	2	0.63
	5	0.54	7	0.28	5	0.58
	7	0.49	5	0.27	7	0.55
MinWFWeight	Zero 0.05	0.68	Zero -0.2	0.34	Zero 0	0.80
	Zero 0.1	0.68	Limit -0.1	0.34	Limit -0.02	0.80
	Zero -0.05	0.68	Limit -0.2	0.34	Zero -0.05	0.80
	Limit -0.01	0.67	Limit -0.5	0.34	Limit -0.01	0.80
	Zero 0.02	0.67	NoLimit	0.34	Zero -0.01	0.80
MinFFreq	NoLimit	0.67	100	0.36	2	0.80
	2	0.67	50	0.36	NoLimit	0.80
	3	0.67	30	0.35	3	0.80
	5	0.67	20	0.35	20	0.80
	7	0.67	15	0.35	15	0.80

Table 6.2. Second-phase performance of a selection of configurations for Spanish on the Moldovan dataset.

Abbrev	Parameter settings							P	S	H
BestCvbmDcEsWiki	**VecSim** Cos		**Weight** Pmi-Tc1Tw3S2P0			**FeatTransf** Weight AftNorm Lb		**0.43**	**0.44**	**0.44**
	DimRed SVD 100	**Smooth** NoSmooth	**VecNorm** L_2	**StopW** true	**MinWFFreq** NoLimit	**MinWFWeight** NoLimit	**MinFFreq** 3			
-	**VecSim** Cos		**Weight** PmiAl-Tc3Tw3S2P0			**FeatTransf** Weight AftNorm Lb		**0.43**	**0.43**	**0.43**
	DimRed SVD 100	**Smooth** NoSmooth	**VecNorm** L_2	**StopW** true	**MinWFFreq** NoLimit	**MinWFWeight** NoLimit	**MinFFreq** 3			
-	**VecSim** Cos		**Weight** Pmi-Tc1Tw3S2P0			**FeatTransf** Weight AftNorm Lb		**0.43**	**0.43**	**0.43**
	DimRed SVD 100	**Smooth** NoSmooth	**VecNorm** L_2	**StopW** true	**MinWFFreq** NoLimit	**MinWFWeight** NoLimit	**MinFFreq** 100			
-	**VecSim** Cos		**Weight** PmiAl-Tc3Tw3S2P0			**FeatTransf** Weight AftNorm Lb		**0.43**	**0.43**	**0.43**
	DimRed SVD 100	**Smooth** NoSmooth	**VecNorm** L_2	**StopW** false	**MinWFFreq** NoLimit	**MinWFWeight** NoLimit	**MinFFreq** 3			
-	**VecSim** Cos		**Weight** Pmi-Tc1Tw3S2P0			**FeatTransf** Weight AftNorm Lb		**0.43**	**0.43**	**0.43**
	DimRed SVD 100	**Smooth** NoSmooth	**VecNorm** L_1	**StopW** true	**MinWFFreq** NoLimit	**MinWFWeight** NoLimit	**MinFFreq** 3			
-	**VecSim** PearsMbAdjCosMod-5.Lb		**Weight** NPmiAl-Tc4Tw4S0P0			**FeatTransf** Weight AftNorm Lb		**0.40**	**0.39**	**0.39**
	DimRed NoDimRed	**Smooth** NoSmooth	**VecNorm** NN	**StopW** true	**MinWFFreq** NoLimit	**MinWFWeight** Zero -0.2	**MinFFreq** NoLimit			
Cos-Pmi	**VecSim** Cos		**Weight** Pmi			**FeatTransf** NoTransf		**0.34**	**0.34**	**0.34**
	DimRed NoDimRed	**Smooth** NoSmooth	**VecNorm** NN	**StopW** false	**MinWFFreq** NoLimit	**MinWFWeight** NoLimit	**MinFFreq** NoLimit			
Cos-PPmi	**VecSim** Cos		**Weight** Pmi			**FeatTransf** NoTransf		**0.34**	**0.33**	**0.33**
	DimRed NoDimRed	**Smooth** NoSmooth	**VecNorm** NN	**StopW** false	**MinWFFreq** NoLimit	**MinWFWeight** Zero 0	**MinFFreq** NoLimit			

Table 6.3. Second-phase performance of a selection of configurations for Hungarian on the second part of the Hungarian TOEFL dataset.

Abbrev	Parameter settings							A
BestCvbmDcHuWiki	**VecSim** MbCosAm		**Weight** NPmiAlpha-Tc4Tw4S0P4			**FeatTransf** Weight AftNorm Sqrt		**0.65**
	DimRed SVD 200	**Smooth** NoSmooth	**VecNorm** L_2	**StopW** false	**MinWFFreq** NoLimit	**MinWFWeight** NoLimit	**MinFFreq** 2	
-	**VecSim** MbCosAm		**Weight** NPmiAlpha-Tc4Tw4S0P4			**FeatTransf** Weight AftNorm Sqrt		**0.65**
	DimRed SVD 200	**Smooth** NoSmooth	**VecNorm** L_1	**StopW** false	**MinWFFreq** NoLimit	**MinWFWeight** NoLimit	**MinFFreq** 2	
-	**VecSim** Cos		**Weight** NPmiAlpha-Tc4Tw4S0P4			**FeatTransf** Weight AftNorm Sqrt		**0.65**
	DimRed SVD 200	**Smooth** NoSmooth	**VecNorm** L_2	**StopW** false	**MinWFFreq** NoLimit	**MinWFWeight** NoLimit	**MinFFreq** 2	
-	**VecSim** Cos		**Weight** NPmiAlpha-Tc4Tw4S0P4			**FeatTransf** Weight AftNorm Sqrt		**0.65**
	DimRed SVD 200	**Smooth** NoSmooth	**VecNorm** L_1	**StopW** false	**MinWFFreq** NoLimit	**MinWFWeight** NoLimit	**MinFFreq** 2	
-	**VecSim** PearsMbMod-1.Lb		**Weight** NPmiAlpha-Tc4Tw4S0P4			**FeatTransf** Weight AftNorm Sqrt		**0.63**
	DimRed SVD 200	**Smooth** NoSmooth	**VecNorm** L_2	**StopW** false	**MinWFFreq** NoLimit	**MinWFWeight** NoLimit	**MinFFreq** 2	
-	**VecSim** PearsMbMod-1.Lb		**Weight** Unis-Tc4Tw4S0P1			**FeatTransf** Weight AftNorm Sqrt		**0.60**
	DimRed NoDimRed	**Smooth** NoSmooth	**VecNorm** L_2	**StopW** false	**MinWFFreq** NoLimit	**MinWFWeight** NoLimit	**MinFFreq** 2	
Cos-PPmi	**VecSim** Cos		**Weight** Pmi			**FeatTransf** NoTransf		**0.53**
	DimRed NoDimRed	**Smooth** NoSmooth	**VecNorm** NN	**StopW** false	**MinWFFreq** NoLimit	**MinWFWeight** Zero 0	**MinFFreq** NoLimit	
Cos-Pmi	**VecSim** Cos		**Weight** Pmi			**FeatTransf** NoTransf		**0.50**
	DimRed NoDimRed	**Smooth** NoSmooth	**VecNorm** NN	**StopW** false	**MinWFFreq** NoLimit	**MinWFWeight** NoLimit	**MinFFreq** NoLimit	

Table 6.4. Results on the test datasets, in descending order of H scores

Lang	Test set	Input data	Configuration	P	S	H
En	MT	DcBnc	BestCvbmDcBnc2	0.71	0.71	0.71
		DcHuWiki	BestCvbmDcHuWiki	0.67	0.68	0.67
		DcEsWiki	BestCvbmDcEsWiki	0.63	0.63	0.63
Es	RG	DcHuWiki	BestCvbmDcHuWiki	0.83	0.83	0.83
		DcBnc	BestCvbmDcBnc2	0.82	0.80	0.81
		DcEsWiki	BestCvbmDcEsWiki	0.80	0.79	0.80
Hu	RG	DcBnc	BestCvbmDcBnc2	0.73	0.72	0.72
		DcEsWiki	BestCvbmDcEsWiki	0.65	0.61	0.63
		DcHuWiki	BestCvbmDcHuWiki	0.58	0.68	0.62

CHAPTER 7

Conclusions

In this thesis we have presented a very detailed and systematic analysis of the possible parameters used during the creation and comparison of feature vectors in distributional semantic models, for English, Spanish and Hungarian, filling a serious research gap. We have identified 10 important parameters of count-vector-based models and 4 relevant ones in case of using semantic vectors as input, and tested numerous settings for all of them. Our analysis included novel parameters and novel parameter settings, and tested all parameters simultaneously, thus also taking the possible interaction between the different parameters into account. To our best knowledge, we are the first to do such a detailed analysis for these parameters, and also to do such an extensive comparison of them across multiple languages.

With our two-step heuristic approach we were searching for the best configurations for all three languages, and were able to find such novel ones, many of them also incorporating novel parameter settings, that significantly outperformed conventional configurations. Although we have used a heuristic approach for the search due to the vast number of possible combinations, we have been able to verify the validity of this approach and the reliability and soundness of its results. Further, we have also verified that a configuration performing well on given input data also works well on other input data of the same type.

In accordance with our intuition, there were several parameters that worked very similarly in case of all three languages. We also found such parameters that were alike for Spanish and Hungarian, and different for English, which we also anticipated. However, it was interesting to see that there was such a parameter that worked similarly for English and Hungarian, but not for Spanish, and we did not find any parameters that worked similarly for the two Indo-European languages, but differently for Hungarian. Although we have found that the very best results are produced by different configurations for the different languages, our cross-language tests showed that all of them work rather well for all languages. Based on this we think that we could find such configurations that are rather language-independent, and give robust and reliable results.

To be able to compare our results with the previous state-of-the-art, we have run such tests where the same data was used as input for both the previous state-of-the-art configurations and our configurations. In case of using raw counts as input and thus being able to optimize all 10 of our examined parameters, our best configurations contained novel parameter settings and clearly outperformed previous state-of-the-art configurations, with a considerable margin in most cases.

When using semantic vectors as input and thus only being able to optimize 4 out of 10 parameters, our best configurations, also incorporating novel parameter settings, performed at least as well as the previous state-of-the-art, with a slight superiority in a couple of cases. Actually, our best model achieved absolute state-of-the-art results compared to all previous models of any type on the most important test datasets. Based on these results we think that our analysis was successful, and we were able to present such new parameter settings and new configurations that are superior to the previous state-of-the-art.

As it could be seen, the size of the input corpus, as well as the used information extraction method greatly influences the results. Therefore we think that doing an analysis similar to our current one for the information extraction phase of DSMs would be a principal direction for future research. Further, in our opinion it would be important to test our proposed new configurations using corpora magnitudes larger than that we could use. It would be even better if our whole heuristic analysis could also be repeated on these huge corpora. Further, although our results seem rather robust and reliable for Spanish and Hungarian too, it would be interesting to redo our analysis on larger and more reliable Spanish and Hungarian datasets, when such datasets will become available in the future.

We think that with this study we significantly contributed to the better understanding of the working and properties of DSMs. Although fully reliable conclusions from our results can only be drawn with respect to DSMs, we think that similar conclusions would hold for other systems based on vector space models too. So in our view our results could also be useful (with some reservations) outside the scope of DSMs, in case of other NLP and non-NLP problems using vector

space models too.

CHAPTER 8

Summary

8.1 Introduction

For many natural language processing (NLP) problems, including information retrieval (Hliaoutakis et al., 2006), spelling correction (Budanitsky and Hirst, 2001) and noun compound interpretation (Dobó and Pulman, 2011) among many others, it is crucial to determine the semantic similarity or semantic relatedness of words. While relatedness takes a wide range of relations between words (including similarity) into account, similarity only considers how much the concepts denoted by the words are truly alike. Thus similarity entices relatedness, but not vice versa. For example, the words "bicycle" and "motorbike" are similar, as both denote 2-wheeled vehicles, and thus they are also related. On the other hand, the

words "postman" and "mail" are highly related, as usually mails are delivered by postmen, and yet they are not similar, as they denote rather different concepts. Further, the words "furnace" and "voyage" are neither similar nor related.

8.1.1 Motivation

Most models are based on the distributional hypothesis of meaning (Harris, 1954), and thus calculate this similarity or relatedness using distributional data extracted from large corpora. These models can be collectively called as distributional semantic models (DSMs) (Baroni and Lenci, 2010; Baroni et al., 2014). In these models first possible features are identified, usually in the form of context words, and then a weight is assigned for each word-feature pair using complex methods, thus creating feature vectors for all words. The similarity or relatedness of words are then calculated by comparing their feature vectors using vector similarity measures. Although DSMs have many possible parameters, a truly comprehensive study of these parameters, also fully considering the dependencies between them, is still missing and would be needed, as also suggested by Levy et al. (2015).

Most papers presenting DSMs focus on only one or two aspects of the problem, and take all the other parameters as granted with some standard setting. For example, the majority of studies simply use cosine as vector similarity measure (e.g. Bruni et al., 2013; Baroni et al., 2014; Speer et al., 2017; Salle et al., 2018) and/or (positive) pointwise mutual information as weighting scheme (e.g. Islam and Inkpen, 2008; Hill et al., 2014b; Salle et al., 2018) out of convention. And even in case of the considered parameters, usually only a handful of possible settings

are tested for. Further, there are also such parameters that are completely ignored by most studies and have not been truly studied in the past, not even separately (e.g. smoothing, vector normalization or minimum feature frequency). What's more, as these parameters can influence each other greatly, evaluating them separately, one-by-one, would not even be sufficient, as that would not account for the interaction between them.

There are a couple of studies that consider several parameters with multiple possible settings, such as Lapesa and Evert (2014) and Kiela and Clark (2014), but even these are far from truly comprehensive, and do not fully test for the interaction between the different parameters. So, although an extensive analysis of the possible parameters and their combinations would be crucial, as also suggested by (Levy et al., 2015), there has been no research to date that would have evaluated these truly comprehensively. Moreover, despite the fact that the best parameter settings for the parameters can differ for different languages, the vast majority of papers consider DSMs for only one language (mostly English), or consider multiple languages but without a real comparison of findings across languages. In this thesis we would like to address these gaps.

8.1.2 Aims and objectives

DSMs have two distinct phases in general. In the first phase statistical information (e.g. raw counts) is extracted from raw data (e.g. a large corpus of raw text), in the form of statistical distributional data. In the second phase, feature vectors are created from the extracted information for each word and these vectors are then compared to each other to calculate the similarity or relatedness of words.

In our study we take the distributional information extracted in the first phase as already granted, and present a systematic study simultaneously testing all important aspects of the creation and comparison of feature vectors in DSMs, also caring for the interaction between the different parameters.

We have chosen to only study the second phase of the DSMs, as the two phases are relatively distinct and independent from each other, and testing for every single possible combination of the parameter settings in the second phase is already unfeasible due to the vast number of combinations. So instead of a full analysis we already had to use a heuristic approach. Thus also trying to test for the parameters of the first phase (e.g. source corpus, context type (window-based or dependency-based) and context size) simultaneously would be unreasonable and unmanageable, and is out of scope of this study. Therefore we have omitted the examination of this phase completely, with one exception to this.

DSMs relying on information extracted from static corpora have two major categories, based on the type of their first phase: count-vector-based (CVBM) and predictive models (PM; also called word embeddings) (Baroni et al., 2014). In order to get a more complete view and due to the huge popularity of predictive models in recent years, in addition to using information extracted from a corpus using a count-vector-based model, we have also done some experiments with information extracted by a predictive model in case of English. Further, later on we also extended our analysis with a model based on semantic vectors constructed from a knowledge graph. Our intuition was that there will be a single configuration that achieves the best results in case of all types of models. However, please note that in the latter case only a part of the considered parameters could be tested for due to the characteristics of such models. That is part of the reason

why we have focused on count-vector-based DSMs more.

During our research we have identified altogether 10 important parameters for the second phase of count-vector-based DSMs, such as vector similarity measures, weighting schemes, feature transformation functions, smoothing and dimensionality reduction techniques. However, only 4 of these parameters are available when predictive or knowledge-graph-based semantic vectors are used as input, as in case of such input the raw counts are not available any more, the weighted vectors are already constructed and their dimensions are usually also reduced.

In the course of our analysis we have simultaneously evaluated each parameter with numerous settings in order to try to find the best possible configuration (configuration) achieving the highest performance on standard test datasets. We have done our extensive analysis for English, Spanish and Hungarian separately, and then we have compared our findings for the different languages.

For some of the tested parameters a large number of possible settings were tested, more than a thousand in some cases, resulting in trillions of possible combinations altogether. While of course also testing the conventionally used parameter settings, we also proposed numerous new variants in case of some parameters. Further, we have tested a vast number of novel configurations, with some of these new configurations considerably outperforming the standard configurations that are conventionally used, and thus achieving state-of-the-art results.

First we have done our analysis for English and evaluated the results extensively (Dobó and Csirik, 2019a). Then we have repeated the same analysis, with an increased number of settings for several parameters, for English, Spanish and Hungarian, and compared the findings across the different languages (Dobó and

Csirik, 2019b).

8.2 Conclusions

In this thesis we have presented a very detailed and systematic analysis of the possible parameters used during the creation and comparison of feature vectors in distributional semantic models, for English, Spanish and Hungarian, filling a serious research gap. We have identified 10 important parameters of count-vector-based models and 4 relevant ones in case of using semantic vectors as input, and tested numerous settings for all of them. Our analysis included novel parameters and novel parameter settings, and tested all parameters simultaneously, thus also taking the possible interaction between the different parameters into account. To our best knowledge, we are the first to do such a detailed analysis for these parameters, and also to do such an extensive comparison of them across multiple languages.

With our two-step heuristic approach we were searching for the best configurations for all three languages, and were able to find such novel ones, many of them also incorporating novel parameter settings, that significantly outperformed conventional configurations. Although we have used a heuristic approach for the search due to the vast number of possible configurations, we have been able to verify the validity of this approach and the reliability and soundness of its results. Further, we have also verified that a configuration performing well on given input data also works well on other input data of the same type.

In accordance with our intuition, there were several parameters that worked very similarly in case of all three languages. We also found such parameters that

were alike for Spanish and Hungarian, and different for English, which we also anticipated. However, it was interesting to see that there was such a parameter that worked similarly for English and Hungarian, but not for Spanish, and we did not find any parameters that worked similarly for the two Indo-European languages, but differently for Hungarian. Although we have found that the very best results are produced by different configurations for the different languages, our cross-language tests showed that all of them work rather well for all languages. Based on this we think that we could find such configurations that are rather language-independent, and give robust and reliable results.

To be able to compare our results with the previous state-of-the-art, we have run such tests where the same data was used as input for both the previous state-of-the-art configurations and our configurations. In case of using raw counts as input and thus being able to optimize all 10 of our examined parameters, our best configurations contained novel parameter settings and clearly outperformed previous state-of-the-art configurations, with a considerable margin in most cases. When using semantic vectors as input and thus only being able to optimize 4 out of 10 parameters, our best configurations, also incorporating novel parameter settings, performed at least as well as the previous state-of-the-art, with a slight superiority in a couple of cases. Actually, our best model achieved absolute state-of-the-art results compared to all previous models of any type on the most important test datasets. Based on these results we think that our analysis was successful, and we were able to present such new parameter settings and new configurations that are superior to the previous state-of-the-art.

As it could be seen, the size of the input corpus, as well as the used information extraction method greatly influences the results. Therefore we think that

doing an analysis similar to our current one for the information extraction phase of DSMs would be a principal direction for future research. Further, in our opinion it would be important to test our proposed new configurations using corpora magnitudes larger than that we could use. It would be even better if our whole heuristic analysis could also be repeated on these huge corpora. Further, although our results seem rather robust and reliable for Spanish and Hungarian too, it would be interesting to redo our analysis on larger and more reliable Spanish and Hungarian datasets, when such datasets will become available in the future.

We think that with this study we significantly contributed to the better understanding of the working and properties of DSMs. Although fully reliable conclusions from our results can only be drawn with respect to DSMs, we think that similar conclusions would hold for other systems based on vector space models too. So in our view our results could also be useful (with some reservations) outside the scope of DSMs, in case of other NLP and non-NLP problems using vector space models too.

9. fejezet

Összefoglalás

9.1. Bevezetés

Számos számítógépes nyelvészeti (NLP) problémához, többek között információ visszakereséshez (Hliaoutakis et al., 2006), helyesírás-javításhoz (Budanitsky and Hirst, 2001) és összetett szó értelmezéshez (Dobó and Pulman, 2011), fontos hogy meg tudjuk határozni szavak szemantikai hasonlóságának vagy kapcsolatának mértékét. Míg a szemantikai kapcsolat számos, szavak között fennálló relációt (többek között a hasonlóságot is) számításba vesz, addig a szemantikai hasonlóság csak a szavak által jelölt fogalmak tényleges egyformaságát veszi figyelembe. Ezáltal a hasonlóságból következik a kapcsolat, de ez fordítva nem igaz. Például, a "bicikli" és a "motorkerékpár" szavak hasonlóak, mivel mindkettő kétkerekű járművet jelöl, így kapcsolódnak is egymáshoz. Ezzel szemben a "postás" és a "levél" szavak közeli kapcsolatban állnak, mivel általában a postás kézbesíti a leveleket, de mégsem hasonlítanak egymásra, mert meglehetősen különböző fogalmakat jelölnek. Továbbá, a "kemence" és a "hajóút" szavak egyáltalán nem hasonlítanak

egymásra és nem is kapcsolódnak egymáshoz.

9.1.1. Motiváció

A legtöbb modell a jelentés eloszlási hipotézisére (Harris, 1954) alapszik, és ezáltal a szemantikai hasonlóság vagy kapcsolat mértékét nagyméretű korpuszból kinyert eloszlási adatok alapján számolja. Ezeket a modelleket gyűjtőnévvel eloszlás alapú szemantikai modelleknek (DSM) szokás nevezni (Baroni and Lenci, 2010; Baroni et al., 2014). Ezekben a modellekben először a lehetséges tulajdonságok kerülnek megállapításra, általában szövegkörnyezeti szavak formájában, ami után a modellek súlyokat rendelnek minden szó-tulajdonság párhoz komplex módszerek segítségével, ezáltal tulajdonság-vektorokat készítve minden szóhoz. A szavak szemantikai hasonlóságának vagy kapcsolatának a mértékét ezt követően a szavak tulajdonság-vektorainak az összehasonlításával számítják ki. Habár a DSM-ek számos lehetséges paraméterrel rendelkeznek, e paraméterek igazán átfogó elemzése, ami a paraméterek egymástól való függését is figyelembe veszi, még hiányzik és szükséges lenne, mint ahogy azt Levy et al. (2015) is sugallja.

A legtöbb DSM-mel foglalkozó kutatás a problémának csak egy vagy két aspektusára fókuszál, és a modell többi paraméterét adottnak veszi valamilyen standard beállítással. Például, a kutatások nagy része megszokásból egyszerűen koszinuszt használ vektorhasonlósági mértékként (pl. Bruni et al., 2013; Baroni et al., 2014; Speer et al., 2017; Salle et al., 2018) és/vagy (pozitív) pontonkénti kölcsönös információt súlyozási sémaként (pl. Islam and Inkpen, 2008; Hill et al., 2014b; Salle et al., 2018). És még a figyelembe vett paraméterek esetén is általá-

ban csak néhány lehetséges beállítást tesztelnek. Továbbá, vannak olyan paraméterek is, amiket a legtöbb tanulmány teljesen figyelmen kívül hagy, és nem is lettek még igazán elemezve a múltban, még külön-külön sem (pl. simítás, vektor-normalizáció vagy a tulajdonságok gyakoriságára minimum limit). Sőt mi több, mivel ezek a paraméterek nagyban befolyásolni tudják egymást, a külön-külön, egyenkénti elemzésük nem is elegendő, mivel az nem veszi figyelembe azok egymásra hatását.

Van néhány olyan kutatás ami több paramétert is tesztel többfajta lehetséges beállítással, mint például Lapesa and Evert (2014) és Kiela and Clark (2014), de ezek is messze vannak attól, hogy igazán átfogó képet adjanak, és szintén nem tesztelik teljes mértékben a különféle paraméterek között fellépő kölcsönhatásokat. Tehát, habár fontos lenne a paramétereket és azok kombinációját részletesen kielemezni, mint ahogy azt Levy et al. (2015) is megemlíti, még mindig nem létezik ezeknek igazán átfogó tanulmánya. Továbbá, annak ellenére, hogy a legjobb paraméter-beállítások a különféle nyelvek esetén különbözőek lehetnek, a tanulmányok döntő többsége általában pusztán egyetlen nyelvvel foglalkozik (legtöbbször az angollal), vagy figyelembe vesz több nyelvet is, de a konklúziók nyelvek közötti részletes összehasonlítása nélkül. Ebben az értekezésben ezeket a kutatási hiányokat szeretnénk betölteni.

9.1.2. Feladat és célkitűzés

A DSM-ek rendszerint két egymástól különálló fázissal rendelkeznek. Az első fázisban statisztikai információt (pl. nyers gyakoriságokat) nyernek ki nyers adatokból (pl. egy nagyméretű nyers szöveges korpuszból), statisztikai eloszlási ada-

tok formájában. A második fázisban tulajdonság-vektorokat készítenek a kinyert információból minden szóhoz, majd ezeket a vektorokat hasonlítják egymáshoz a szavak hasonlósági vagy kapcsolati mértékének a megállapításához. Mi a kutatásunk során az első fázisban kinyert információt már adottnak vesszük, és egy szisztematikus, párhuzamos elemzését végezzük el a tulajdonságvektorok készítése és összehasonlítása során használt tulajdonságoknak, miközben a tulajdonságok egymásra hatását is figyelembe vesszük.

Azért döntöttünk úgy, hogy csak a DSM-ek második fázisát elemezzük, mivel a két fázis egymástól meglehetősen különálló és független, és a második fázis minden egyes lehetséges paraméter-érték kombinációjának tesztelése már így is lehetetlen a lehetséges kombinációk óriási száma miatt. Ezért egy teljes analízis helyett már így is egy heurisztikus módszert kellett alkalmaztunk. Tehát ezen felül még az első fázis különféle paramétereit (pl. használt korpusz, szövegkörnyezeti típus (ablak-alapú vagy dependencia-alapú) és szövegkörnyezeti méret) is tesztelni ésszerűtlennek és megvalósíthatatlannak tűnt, és így e kutatás hatókörén kívülre esett. Ezért ennek a fázisnak a vizsgálatát teljes egészében kihagytuk, egy kivétellel.

A statikus korpuszokból kinyert információkon alapuló DSM-eknek két jelentős csoportja van az első fázisuk alapján: gyakorisági-vektor-alapú (CVBM) és prediktív modellek (PM; más névvel szóbeágyazási modellek) (Baroni et al., 2014). A prediktív modellek elmúlt évekbeli nagy népszerűsége miatt, továbbá azért, hogy még teljesebb képet kapjunk, a gyakorisági-vektor-alapú modellek által korpuszokból kinyert információk mellett elvégeztünk néhány kísérletet prediktív modellek által kinyert információkkal is az angol nyelv esetén. Továbbá, a későbbiekben még kiegészítettük az elemzésünket egy olyan modellel is, ami

tudás-gráfból kinyert szemantikai vektorokon alapszik. A megérzésünk az volt, hogy lesz egy olyan konfiguráció ami a legjobb eredményt fogja elérni mindhárom típusú modell esetén. Azt azonban meg kell jegyezzük, hogy a prediktív modellek esetén a figyelembe vett paramétereknek csak egy részét lehetett tesztelni e modellek jellegzetességei miatt. Részben ezért is fókuszáltunk inkább a gyakorisági-vektor-alapú modellekre.

A kutatásunk során összességében 10 fontos paramétert azonosítottunk a gyakorisági-vektor-alapú DSM-ek második fázisában, mint például a vektorhasonlósági mértéket, a súlyozási sémát, a tulajdonság-transzformációt, a simítást és a dimenzió-csökkentést. Ezek közül azonban összesen 4 érhető el prediktív illetve tudás-gráf-alapú szemantikai vektorok használata esetén, mivel ilyen inputok használatakor a nyers gyakoriságok már nem érhetőek el, a súlyozott vektorok már elkészültek és általában már a dimenzió-csökkentés is végrehajtásra került rajtuk.

Elemzésünk során e paramétereket párhuzamosan értékeltük ki számos beállítással annak érdekében, hogy megtaláljuk a legjobb konfigurációt, amit a lehető legmagasabb pontszámokat éri el a standard tesztadatbázisokon. Az átfogó elemzésünket angolra, spanyolra és magyarra külön-külön is megcsináltuk, majd a különböző nyelvek esetén levont konklúziókat összehasonlítottuk.

Néhány paraméterre nagy mennyiségű, akár több ezer lehetséges beállítást is teszteltünk, ami több milliárd lehetséges paraméter-beállítási kombinációt eredményezett. Amellett, hogy természetesen minden paraméter konvencionálisan alkalmazott beállítását is teszteltük, számos új variánst javasoltunk mi is. Továbbá, számos új konfigurációt teszteltünk, amik közül némelyek az általánosan használt, standard konfigurációknál messze jobb eredményt érnek el, és az eddig

ismert legjobb konfigurációknál is jobb eredményeket érnek el.

Első körben az elemzésünket angolra végeztük el, és az eredményeket azon elemeztük ki részletesen (Dobó and Csirik, 2019a). Ezt követően megismételtük ugyanezt az elemzést, néhány paraméter esetén bővített beállítási opciókkal, angolra, spanyolra és magyarra is, és a különböző nyelvekre levont konklúziókat összevetettük egymással (Dobó and Csirik, 2019b).

9.2. Konklúziók

Az értekezésben az eloszlás alapú szemantikai modellek tulajdonságvektorainak készítése és összehasonlítása során használt paramétereknek egy nagyon részletes és szisztematikus elemzését prezentáltuk angolra, spanyolra és magyarra, amivel egy komoly kutatási hiányt töltöttünk be. Gyakorisági-vektor-alapú modellek esetén 10, míg prediktív és tudás-gráf-alapú modellek esetén 4 fontos paramétert azonosítottunk, és ezek mindegyikéhez számos beállítást teszteltünk. Az elemzésünk során teszteltünk új paramétereket és új paraméter-beállításokat, továbbá minden paramétert párhuzamosan vizsgáltunk, ezáltal ezek esetleges egymásra hatását is figyelembe véve. Tudomásunk szerint mi voltunk az elsők, akik e paraméterek ilyen részletes elemzését elvégezték, és szintén mi voltunk az elsők, akik a különböző nyelvek esetén levont konklúziókat részletesen összehasonlították.

A két lépéses heurisztikus módszerünk segítségével mindhárom nyelvre megkerestük a legjobb konfigurációt, ami során olyan konfigurációkat találtunk, amik egy része új paraméter-beállításokat is tartalmaz, amik lényegesen jobb eredményt érnek el az általánosan használt beállításoknál. Habár egy heurisztikus

módszert használtunk a kereséshez a lehetséges konfigurációk óriási száma miatt, igazolni tudtuk e módszerünk helyességét és az általa adott eredmények megbízhatóságát és helyességét. Továbbá igazolni tudtuk azt is, hogy egy adott bemeneti adattípuson jól működő konfiguráció másik azonos típusú bemenet használata esetén is jól működik.

A kezdeti sejtésünknek megfelelően volt jó néhány olyan paraméter, ami mindhárom nyelv esetén nagyon hasonlóan működött. Találtunk olyan paramétereket is, amik spanyolra és magyarra hasonlóan működnek, de angolra másképp, ami szintén várható volt. Mindemellett meglepődve tapasztaltuk azt, hogy volt olyan paraméter is, ami angolra és magyarra hasonlóan működött, míg spanyolra máshogyan, illetve nem találtunk olyan paramétert, amit a két indoeurópai nyelvre azonosan működött volna, de magyarra másképp. Habár azt tapasztaltuk, hogy a legjobb eredményt a különböző nyelvek esetén különböző konfigurációval lehet elérni, a nyelvek közötti tesztjeink megmutatták azt, hogy ezek mindegyike meglehetősen jól működik mindhárom nyelv esetén. Ez alapján úgy gondoljuk, hogy sikerült olyan konfigurációkat találni, melyek meglehetősen nyelv-függetlenek, és robosztus és megbízható eredményeket adnak.

Annak érdekében, hogy az eredményeinket össze tudjuk hasonlítani az eddig ismert legjobb módszerek eredményeivel, olyan teszteket is futtattunk, amiben azonos bemeneti adatokat használtunk a jelenleg ismert legjobb konfigurációkhoz és a mi konfigurációinkhoz is. Nyers frekvenciák bemenetként való használata esetén, amikor mind a 10 paramétert tudtunk vizsgálni, a legjobb konfigurációink tartalmaztak új paraméter-beállításokat és a eddigi legjobb konfigurációknál egyértelműen jobb eredményeket értek el, általában számottevően magasabb pontszámokkal. Szemantikus vektorok bemenetként való használata esetén,

amikor a 10-ből csak 4 paramétert tudtunk vizsgálni, a legjobb konfigurációink legalább olyan jól teljesítettek, mint a eddigi legjobb konfigurációk, és néhány esetben kis fölénnyel is rendelkeztek. Igazából a legjobb modellünk abszolút legjobb eredményt ért el, minden eddigi modellnél jobban teljesítve a legfontosabb tesztadathalmazokon. Ezek alapján úgy gondoljuk, hogy az elemzésünk sikeres volt, és sikerült olyan új paraméter-beállításokat és új konfigurációkat bemutatni, amik az eddig ismertnél jobb eredményeket tudnak elérni, és ezáltal túlszárnyalják az eddig ismert legjobb konfigurációkat.

Ahogy a tesztek során látszódott, a bemenetként használt korpusz, illetve az alkalmazott információ-kinyerési módszer nagyban befolyásolja az eredményeket. Ebből kifolyólag úgy gondoljuk, egy a mostanihoz hasonló elemzés elvégzése a DSM-ek információkinyerési fázisán kulcsfontosságú iránya lehetne a jövőbeni kutatásoknak. Továbbá, véleményünk szerint fontos lenne az általunk újonnan javasolt konfigurációkat az általunk használt szöveges korpuszoknál nagyságrendekkel nagyobbakon tesztelni. Ennél még jobb lenne, ha ezeken az óriási korpuszokon a teljes elemzést meg lehetne ismételni. Ezen felül, habár az eredményeink meglehetősen robosztusnak és megbízhatónak tűnnek spanyolra és magyarra is, érdekes lenne az elemzésünket megismételni nagyobb és megbízhatóbb spanyol és magyar tesztadatbázisokon is, amint ilyen adathalmazok elérhetővé válnak.

Úgy gondoljuk, hogy e tanulmánnyal nagyban hozzájárultunk a DSM-ek működésének és tulajdonságainak a megértéséhez. Habár az eredményeinkből teljesen megbízható konklúziókat csak DSM-ekre tekintettel tudunk levonni, úgy gondoljuk, hogy hasonló konklúziók érvényesek lennének más, szintén vektortér modelleken alapuló rendszerek esetén is. Ezért úgy gondoljuk, hogy az ered-

ményeink hasznosak lehetnek (némi fenntartással) a DSM-ek tárgykörén kívül is, más, vektor-tér modelleket alkalmazó számítógépes nyelvészeti vagy tetszőleges egyéb probléma esetén is.

References

John L Ackrill. *Categories and De interpretatione*. Oxford University Press, 1975.

Mohamed Ben Aouicha, Mohamed Ali Hadj Taieb, and Abdelmajid Ben Hamadou. Taxonomy-based information content and wordnet-wiktionary-wikipedia glosses for semantic relatedness. *Applied Intelligence*, 45(2):475–511, 2016.

Mark Aronoff and Janie Rees-Miller. *The handbook of linguistics*. Blackwell Publishers Ltd, 2003.

Rajendra Banjade, Nabin Maharjan, Nobal B Niraula, Vasile Rus, and Dipesh Gautam. Lemon and Tea Are Not Similar: Measuring Word-to-Word Similarity by Combining Different Methods. In *Computational Linguistics and Intelligent Text Processing*, pages 335–346. Springer, 2015.

Marco Baroni and Alessandro Lenci. Distributional Memory: A General Framework for Corpus-Based Semantics. *Computational Linguistics*, 36(4):673–721, 2010.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. The waCky wide web: A collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226, 2009.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *52nd ACL*, pages 238–247, 2014.

BNC Consortium. The British National Corpus, version 2 (BNC World), 2001.

Laurel J Brinton. *The structure of modern English: A linguistic introduction*. John Benjamins Publishing, 2000.

Elia Bruni, Nam Khan Tran, and Marco Baroni. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 48:1–47, 2013.

Alexander Budanitsky and Graeme Hirst. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and other lexical resources*, page 2, 2001.

John A Bullinaria and Joseph P Levy. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39(3):510–526, aug 2007.

John A Bullinaria and Joseph P Levy. Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and SVD. *Behavior research methods*, 44(3): 890–907, 2012.

Myles Burnyeat et al. *The Theaetetus of Plato*. Hackett Publishing, 1990.

José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. A Framework for the Construction of Monolingual and Cross-lingual Word Similarity Datasets. In *53rd ACL*, pages 1–7, 2015.

Jose Camacho-Collados, Mohammad Taher Pilehvar, Nigel Collier, and Roberto Navigli. Semeval-2017 task 2: Multilingual and cross-lingual semantic word similarity. In *SemEval-2017*, pages 15–26, 2017.

Sung-hyuk Cha. Comprehensive Survey on Distance / Similarity Measures between Probability Density Functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, 1(4):300–307, 2007.

Qianqian Chen. *Representing semantic relatedness*. PhD thesis, University of Technology, Sydney, 2016.

Stanley Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 13:359–394, 1999a.

Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–393, 1999b.

Alexey Chernyak. Meaning as convention: a puzzling effect of aristotelian theory of names. *Schole*, 11(1):78–94, 2017.

Fenia Christopoulou, Eleftheria Briakou, Elias Iosif, and Alexandros Potamianos. Mixture of Topic-Based Distributional Semantic and Affective Models. In *12th IEEE ICSC*, pages 203–210, 2018.

Yu-Ming Chu and Shou-Wei Hou. Sharp bounds for Seiffert mean in terms of contraharmonic mean. In *Abstract and Applied Analysis*. Hindawi, 2012.

Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, 1990. ISSN 08912017.

R L Cilibrasi and P M B Vitányi. The google similarity distance. *IEEE Transactions on knowledge and data engineering*, pages 370–383, 2007.

Jacob Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

Guillem Collell, Ted Zhang, and Marie-francine Moens. Imagined Visual Representations as Multimodal Embeddings. *Proceedings of the 31th Conference on Artificial Intelligence (AAAI 2017)*, pages 4378–4384, 2017.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing. *Proceedings of the 25th international conference on Machine learning - ICML '08*, 20(1): 160–167, 2008.

James Richard Curran. From Distributional to Semantic Similarity. page 177, 2004.

Ido Dagan, Shaul Marcus, and Shaul Markovitch. Contextual word similarity and estimation from sparse data. *Computer Speech & Language*, 9(2):123–152, apr 1995.

Simon De Deyne, Amy Perfors, and Daniel J. Navarro. Predicting human similarity judgments with distributional models: The value of word associations. In *26th IJCAI*, pages 4806–4810, 2017.

Michel Marie Deza and Elena Deza. *Encyclopedia of distances*. Springer-Verlag Berlin Heidelberg, 2016.

Georgiana Dinu. *Word meaning in context: A probabilistic model and its application to Question Answering*. PhD thesis, Universität des Saarlandes, 2011.

András Dobó. Multi-D Kneser-Ney Smoothing Preserving the Original Marginal Distributions. *Research in Computing Science*, 147(6):11–25, 2018.

András Dobó and János Csirik. Magyar és angol szavak szemantikai hasonlóságának automatikus kiszámítása. In *IX. Magyar Számítógépes Nyelvészeti Konferencia*, pages 213–224, 2012.

András Dobó and János Csirik. Computing semantic similarity using large static corpora. In *SOFSEM 2013: Theory and Practice of Computer Science. LNCS 7741*, pages 491–502, 2013.

András Dobó and János Csirik. A comprehensive study of the parameters in the creation and comparison of feature vectors in distributional semantic models. *Journal of Quantitative Linguistics*, 2019a.

András Dobó and János Csirik. Comparison of the best parameter settings in the creation and comparison of feature vectors in distributional semantic models across multiple languages. In *15th International Conference on Artificial Intelligence Applications and Innovations. IFIP AICT (in press)*, 2019b.

András Dobó and Stephen G Pulman. Interpreting noun compounds using paraphrases. *Procesamiento del Lenguaje Natural*, 46:59–66, 2011.

Stefan Evert. *The Statistics of Word Cooccurrences Word Pairs and Collocations*. PhD thesis, Universität Stuttgart, 2005.

Stefan Evert. Corpora and collocations. *Corpus Linguistics. An International Handbook*, pages 1–53, 2008.

Manaal Faruqui and Chris Dyer. Non-distributional Word Vector Representations. pages 464–469, 2015.

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah A Smith. Sparse Overcomplete Word Vector Representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1491–1500, 2015.

C Fellbaum. *WordNet: An electronic lexical database*. The MIT Press, Cambridge, MA, 1998.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: the concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131, 2002.

John R Firth. A synopsis of linguistic theory, 1930-1955. In *Studies in linguistic analysis*, pages 1–32. Blackwell, 1957.

Gottlob Frege. *Die Grundlagen der Arithmetik: eine logisch mathematische Untersuchung über den Begriff der Zahl*. W. Koebner, 1884.

Pascale Fung and Kathleen McKeown. A Technical Word and Term Translation Aid using Noisy Parallel Corpora Across Language Groups. *Machine Translation*, 12, 1997.

Joshua T Goodman. A bit of progress in language modeling. *Computer Speech & Language*, 15(4):403–434, 2001.

Ulrike Hahn, Nick Chater, and Lucy B Richardson. Similarity as transformation. *Cognition*, 87(1):1–32, 2003.

Sébastien Harispe, Sylvie Ranwez, Stefan Janaqi, and Jacky Montmain. Semantic similarity from natural language and ontology analysis. *Synthesis Lectures on Human Language Technologies*, 8(1):1–254, 2015.

Zellig S. Harris. Distributional Structure. *WORD*, 10(2-3):146–162, 1954.

Samer Hassan and Rada Mihalcea. Cross-lingual semantic relatedness using encyclopedic knowledge. In *2009 CoNLL*, pages 1192–1201, 2009.

Samer Hassan and Rada Mihalcea. Semantic Relatedness Using Salient Semantic Analysis. *Proceedings of the 25th AAAI Conference on Artificial Intelligence, (AAAI 2011)*, pages 884–889, 2011.

Priska Herger. *Learning semantic relations with distributional similarity*. PhD thesis, Master's thesis, TU Berlin, 2014.

Felix Hill, KyungHyun Cho, Sebastien Jean, Coline Devin, and Yoshua Bengio. Not All Neural Embeddings are Born Equal. pages 1–5, 2014a.

Felix Hill, Roi Reichart, and Anna Korhonen. Multi-Modal Models for Concrete and Abstract Concept Meaning. *Transactions of the Association for Computational Linguistics*, 2:285–296, 2014b.

Felix Hill, Roi Reichart, and Anna Korhonen. SimLex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation. *Computational Linguistics*, 41(4):665–695, 2015.

Angelos Hliaoutakis, Giannis Varelas, Epimenidis Voutsakis, Euripides G M Petrakis, and Evangelos Milios. Information retrieval by semantic similarity. *International journal on semantic Web and information systems*, 2(3):55–73, 2006.

Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. Improving-Word Representations via Global Context and MultipleWord Prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, number July, pages 873–882, 2012.

Thad Hughes and Daniel Ramage. Lexical Semantic Relatedness with Random Graph Walks. *Computational Linguistics*, 7(June):581–589, 2007.

Elias Iosif and Alexandros Potamianos. Similarity computation using semantic networks created from web-harvested data. *Natural Language Engineering*, 21(1):49–79, 2015.

Elias Iosif, Spiros Georgiladakis, and Alexandros Potamianos. Cognitively Motivated Distributional Representations of Meaning. In *10th Language Resources and Evaluation Conference*, 2016.

Aminul Islam and Diana Inkpen. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data*, 2(2): 1–25, 2008.

Shahida Jabeen. *Exploiting wikipedia semantics for computing word associations*. PhD thesis, Victoria University of Wellington, 2014.

Theo MV Janssen. Frege, contextuality and compositionality. *Journal of Logic, Language and Information*, 10(1):115–136, 2001.

William P. Jones and George W. Furnas. Pictures of relevance: A geometric analysis of similarity measures. *Journal of the American Society for Information Science*, 1987.

D Jurafsky and J H Martin. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson Education, Upper Saddle River, NJ, second edition, 2009.

Jerrold J Katz and Jerry A Fodor. The structure of a semantic theory. *Language*, 39(2): 170–210, 1963.

Douwe Kiela and Stephen Clark. A systematic study of semantic vector space model parameters. In *2nd CVSC at EACL*, pages 21–30, 2014.

Adam Kilgarriff. Googleology is Bad Science. *Computational Linguistics*, 33(1):147–151, 2007.

Reinhard Kneser and Hermann Ney. Improved backing-off for m-gram language modeling. In *IEEE ICASSP 1995*, pages 181–184, 1995a.

Reinhard Kneser and Hermann Ney. Improved backing-off for m-gram language modeling. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, pages 181–184, 1995b.

András Kornai. The algebra of lexical semantics. In *the Mathematics of Language*, pages 174–199. Springer, 2010.

András Kornai. *Semantics*. Springer International Publishing (in press), 2019.

Swarnim Kulkarni and Doina Caragea. Computation of the Semantic Relatedness between Words using Concept Clouds. In *In: International Conference on Knowledge Discovery and Information Retrieval*, pages 183–188, Setubal, 2009. INSTICC Press.

T K Landauer and S T Dumais. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2):211–240, 1997.

Ronald W Langacker. *Cognitive grammar: A basic introduction*. Oxford University Press, 2008.

Gabriella Lapesa and Stefan Evert. A Large Scale Evaluation of Distributional Semantic Models: Parameters , Interactions and Model Selection. *Transactions of the Association for Computational Linguistics*, 2:531–545, 2014.

Angeliki Lazaridou, Nghia The Pham, and Marco Baroni. Combining Language and

Vision with a Multimodal Skip-gram Model. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 153–163, 2015.

Yang-Yin Lee, Hao Ke, Hen-Hsen Huang, and Hsin-Hsi Chen. Combining Word Embedding and Lexical Database for Semantic Relatedness Measurement. In *Proceedings of the 25th International Conference Companion on World Wide Web - WWW '16 Companion*, pages 73–74, 2016.

Omer Levy, Yoav Goldberg, and Ido Dagan. Improving Distributional Similarity with Lessons Learned from Word Embeddings. *Transactions of the ACL*, 3:211–225, 2015.

D Lin. An information-theoretic definition of similarity. In *In: 15th International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann Publishers Inc., San Francisco, 1998a.

Dekang Lin. Extracting Collocations from Text Corpora. *First Workshop on Computational Terminology*, pages 57–63, 1998b.

Will Lowe. Towards a Theory of Semantic Space. *Proceedings of the Twenty-Third Annual Conference of the Cognitive Science Society*, pages 576–581, 2001.

C D Manning and H Schütze. *Foundations of statistical natural language processing*. MIT Press, Cambridge, 1999.

Oren Melamud, Omer Levy, and Ido Dagan. A Simple Word Embedding Model for Lexical Substitution. In *1st CVSC at NAACL*, pages 1–7, 2015.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. *Conference on Neural Information Processing Systems (NIPS)*, pages 1–9, 2013a.

Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In *ICLR 2013*, pages 1–12, 2013b.

Cornelia D Moldovan, Pilar Ferré, Josep Demestre, and Rosa Sánchez-Casas. Semantic similarity: normative ratings for 185 Spanish noun triplets. *Behavior research methods*, 47(3):788–799, 2015.

Nikola Mrkšić, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young. Semantic Specialization of Distributional Word Vector Spaces using Monolingual and Cross-Lingual Constraints. *Transactions of the Association of Computational Linguistics*, 5(1):309–324, 2017.

Preslav Nakov. Using the Web as an Implicit Training Set : Application to Noun Compound Syntax and Semantics. pages 1–405, 2007.

Hermann Ney and Ute Essen. On smoothing techniques for bigram-based natural language modelling. In *1991 International Conference on Acoustics, Speech, and Signal Processing*, pages 825–828, 1991.

Attila Novák and Borbála Novák. Magyar szóbeágyazási modellek kézi kiértékelése. In *XIV. Magyar Számítógépes Nyelvészeti Konferencia*, pages 66–77, 2018.

Charles Kay Ogden and Ivor Armstrong Richards. *The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism*. K. Paul, Trench, Trubner & Company, Limited, 1923.

Patrick Pantel and Dekang Lin. Discovering word senses from text. *8th ACM SIGKDD*, 41:613, 2002.

Adrián Jiménez Pascual and Sumio Fujita. Text similarity function based on word embeddings for short text analysis. In *18th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing 2017)*, 2017.

Pavel Pecina. Lexical association measures and collocation extraction. *Language Resources and Evaluation*, 44(1-2):137–158, 2010.

Francis Jeffry Pelletier. Did frege believe frege's principle? *Journal of Logic, Language and information*, 10(1):87–114, 2001.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe : Global Vectors for Word Representation. In *EMNLP 2014*, pages 1532–1543, 2014.

Mohammad Taher Pilehvar and Roberto Navigli. From senses to texts: An all-in-one graph-based approach for measuring semantic similarity. *Artificial Intelligence*, 228: 95–128, 2015.

Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli. Align , Disambiguate and Walk: A Unified Approach for Measuring Semantic Similarity. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1341–1351, 2013.

Reinhard Rapp. Word Sense Discovery Based on Sense Descriptor Dissimilarity. In *In: 9th Machine Translation Summit*, volume 33, pages 315–322. Association for Machine Translation in the Americas, Stroudsburg, 2003.

Gábor Recski, Eszter Iklódi, Katalin Pajkossy, and András Kornai. Measuring semantic similarity of words using concept networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 193–200, 2016.

Joel W Reed, Jiao Yu, Thomas E Potok, Brian A Klump, Mark T Elmore, and Ali R Hurson. TF-ICF: A new term weighting scheme for clustering dynamic data streams. *Proceedings - 5th International Conference on Machine Learning and Applications, ICMLA 2006*, pages 258–263, 2006.

Samuel Reese, Gemma Boleda, Montse Cuadros, Lluís Padró, and German Rigau. Wikicorpus: A Word-Sense Disambiguated Multilingual Wikipedia Corpus. In *7th LREC*, pages 1418–1421, 2010.

Alfréd Rényi. On measures of entropy and information. *Entropy*, 547(c):547–561, 1961.

Philip Resnik. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *In: 14th International Joint Conference on Artificial Intelligence*, pages 448–453. Morgan Kaufmann Publishers Inc., San Francisco, nov 1995.

Martin Riedl. *Unsupervised Methods for Learning and Using Semantics of Natural Language*. PhD thesis, Technischen Universität Darmstadt, 2016.

Sascha Rothe and Hinrich Schütze. AutoExtend: Combining Word Embeddings with Semantic Resources. *Computational Linguistics*, pages 1–25, 2017.

Herbert Rubenstein and John B. Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, 1965.

Alexandre Salle, Marco Idiart, and Aline Villavicencio. Matrix Factorization using Window Sampling and Negative Sampling for Improved Word Representations. In *54th ACL*, page 419, 2016a.

Alexandre Salle, Marco Idiart, and Aline Villavicencio. Enhancing the LexVec Distributed Word Representation Model Using Positional Contexts and External Memory. *arXiv preprint arXiv:1606.01283*, 2016b.

Alexandre Salle, Marco Idiart, and Aline Villavicencio. LexVec, 2018. URL `https://github.com/alexandres/lexvec/blob/master/README.md`. [Accessed 01.04.2019].

Enrico Santus, Emmanuele Chersoni, Alessandro Lenci, Chu-Ren Huang, and Philippe Blache. Testing APSyn against Vector Cosine on Similarity Estimation. In *30th PACLIC*, pages 229–238, 2016.

Enrico Santus, Hongmin Wang, Emmanuele Chersoni, and Yue Zhang. A Rank-Based Similarity Metric for Word Embeddings. *arXiv preprint arXiv:1805.01923*, 2018.

Ferdinand de Saussure. *Cours de linguistique générale*. Payot, 1916.

Louis L Scharf and Cédric Demeure. *Statistical signal processing: detection, estimation, and time series analysis*. Addison-Wesley Reading, MA, 1991.

William A. Scott. Reliability of Content Analysis: The Case of Nominal Scale Coding. *Public Opinion Quarterly*, 19(3):321, 1955.

Walid Shalaby and Wlodek Zadrozny. Measuring Semantic Relatedness using Mined Semantic Analysis. *arXiv preprint arXiv:1512.03465*, 2016.

Roger N Shepard. The analysis of proximities: multidimensional scaling with an unknown distance function. i. *Psychometrika*, 27(2):125–140, 1962.

Christopher L. Sistrom and Cynthia W. Garvan. Proportions, Odds, and Risk. *Radiology*, 230(1):12–19, 2004.

Robert Speer, Joshua Chin, and Catherine Havasi. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *31st AAAI*, pages 4444–4451, 2017.

Peter D Turney and Patrick Pantel. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, 37:141–188, 2010.

Amos Tversky. Features of similarity. *Psychological review*, 84(4):327–352, 1977.

Maksym Vakulenko. Calculation of Semantic Distances Between Words: From Synonymy to Antonymy. *Journal of Quantitative Linguistics*, pages 1–13, 2018.

Vladimir Vapnik, Steven E Golowich, and Alex Smola. Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing. *Advances in Neural Information Processing Systems*, 9:281–287, 1997.

Ivan Vulić, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. Hyperlex: A large-scale evaluation of graded lexical entailment. *Computational Linguistics*, 43(4): 781–835, 2017.

Julie Weeds and David Weir. Co-occurrence Retrieval: A Fexible Framework for Lexical Distributional Similarity. *Computational Linguistics*, 31(4):339–445, 2005.

Julie Elizabeth Weeds. *Measures and Applications of Lexical Distributional Similarity*. PhD thesis, University of Sussex, 2003.

David Weir, Julie Weeds, Jeremy Reffin, and Thomas Kober. Aligning Packed Dependency Trees: A Theory of Composition for Distributional Semantics. *Computational Linguistics*, 42(4):727–761, 2016.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Charagram: Embedding Words and Sentences via Character n-grams. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1504–1515, 2016.

Thomas H Wonnacott and Ronald J Wonnacott. *Introductory statistics*. Wiley New York, 1990.

Wilhelm Max Wundt. *Grundriss der psychologie*. A. Kröner, 1920.

Wen-tau Yih and Ann Arbor. Measuring Word Relatedness Using Heterogeneous Vector Space Models University of Michigan. *2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (2010):616–620, 2012.

Wenpeng Yin and Hinrich Schütze. Learning Word Meta-Embeddings. In *ACL*, pages 1351–1360, 2016.

Hui Zhang and David Chiang. Kneser-ney smoothing on expected counts. In *The 52nd Annual Meeting of the Association for Computational Linguistics*, pages 765–774, 2014.

K Zhang, K Zhu, and S Hwang. An Association Network for Computing Semantic Relatedness. *Twenty-Ninth AAAI Conference on . . .* , pages 593–599, 2015.

Lei Zhang, Qi-ming Zhang, Yi-guo Wang, and Dong-lin Yu. Selecting an appropriate interestingness measure to evaluate the correlation between syndrome elements and symptoms. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (LNCS 7104)*, pages 372–383. Springer, 2011.

Appendices

APPENDIX A

A list of the most important vector similarity measures tested

As most distance and similarity measures are not defined in case at least one of the vectors to be compared are of zero length, in such cases we have always taken the similarity score to be 0. Further, to reduce the unnecessary special cases in the formulas, we have used the following two simplifications: $\frac{0}{0} = 0$ and $0 * ln(0) = 0$.

	Definition	Reference
Cos	$s(u,v) = \frac{\sum_{i=1}^{n} u_i * v_i}{\sqrt{\sum_{i=1}^{n} u_i^2} * \sqrt{\sum_{i=1}^{n} v_i^2}}$	(Jones and Furnas, 1987)
AdjCos	$s(u,v) = \begin{cases} 1, & Cos(u,v) \geq \lambda \\ \frac{Cos(u,v)}{\lambda}, & otherwise \end{cases}$ $\lambda = 0.1$	(Shalaby and Zadrozny, 2016)
AdjCosPFMod	$s(u,v) = \begin{cases} 1, & PFMod(u,v) \geq \lambda \\ \frac{PFMod(u,v)}{\lambda}, & otherwise \end{cases}$ $\lambda = 1$	
ApSyn	$s(u,v) = \sum_{i=1}^{n} \begin{cases} \frac{2}{rank(u_i)+rank(v_i)}, & u_i \neq 0 \wedge v_i \neq 0 \\ 0, & otherwise \end{cases}$ vectors are sorted and reduced according to the original method	(Santus et al., 2016)
ApSynP	$s(u,v) = \sum_{i=1}^{n} \begin{cases} \frac{2}{rank(u_i)^p+rank(v_i)^p}, & u_i \neq 0 \wedge v_i \neq 0 \\ 0, & otherwise \end{cases}$ $p = 0.1$ vectors are sorted and reduced according to the original method	(Santus et al., 2018)
$AvgL_1L_{Inf}$		(Cha, 2007)
Canberra	$d(u,v) = \sum_{i=1}^{n} \frac{\lvert u_i - v_i \rvert}{\lvert u_i \rvert + \lvert v_i \rvert}$	(Cha, 2007)
ChenCorr	$s(u,v) = \frac{1}{n} \sum_{i=1}^{n} \frac{u_i * v_i}{u_i + v_i - u_i * v_i}$	(Chen, 2016)
ContraHMeanMod	$s(u,v) = \sum_{i=1}^{n} \begin{cases} \frac{u_i^2+v_i^2}{\lvert u_i \rvert + \lvert v_i \rvert}, & \lvert u_i \rvert + \lvert v_i \rvert \neq 0 \\ 0, & otherwise \end{cases}$	based on (Chu and Hou, 2012)
DFVMB	$d(u,v) = \sqrt{\frac{\sum_{i=1}^{n}((u_i - \bar{u})^2 + (v_i - \bar{v}))^2}{2 * n}}$	inspired by StdLike
Dice-1	$s(u,v) = \frac{2 * \sum_{i=1}^{n} min(u_i, v_i)}{\sum_{i=1}^{n} u_i + \sum_{i=1}^{n} v_i}$	(Kiela and Clark, 2014)
Dice-1Mod	$s(u,v) = \frac{2 * \sum_{i=1}^{n} min(u_i, v_i)}{\sum_{i=1}^{n} \lvert u_i \rvert + \sum_{i=1}^{n} \lvert v_i \rvert}$	based on (Kiela and Clark, 2014)
Dice-2	$s(u,v) = \frac{2 * \sum_{i=1}^{n} u_i * v_i}{\sum_{i=1}^{n} u_i^2 + \sum_{i=1}^{n} v_i^2}$	(Cha, 2007)
Jaccard1	$s(u,v) = \frac{\sum_{i=1}^{n} u_i * v_i}{\sum_{i=1}^{n} u_i + \sum_{i=1}^{n} v_i}$	(Kiela and Clark, 2014)
Jaccard1Mod	$s(u,v) = \frac{\sum_{i=1}^{n} u_i * v_i}{\sum_{i=1}^{n} \lvert u_i \rvert + \sum_{i=1}^{n} \lvert v_i \rvert}$	based on (Kiela and Clark, 2014)
Kulczynski	$s(u,v) = \frac{\sum_{i=1}^{n} min(u_i, v_i)}{\sum_{i=1}^{n} \lvert u_i - v_i \rvert}$	(Deza and Deza, 2016)

	Definition	Reference
$L_{0.5}$	$d(u,v) = \left(\sum_{i=1}^{n} \lvert u_i - v_i \rvert^{\frac{1}{2}}\right)^2$	(Cha, 2007)
L_1	$d(u,v) = \sum_{i=1}^{n} \lvert u_i - v_i \rvert$	(Cha, 2007)
L_2	$d(u,v) = \left(\sum_{i=1}^{n} \lvert u_i - v_i \rvert^2\right)^{\frac{1}{2}}$	(Cha, 2007)
L_3	$d(u,v) = \left(\sum_{i=1}^{n} \lvert u_i - v_i \rvert^3\right)^{\frac{1}{3}}$	(Cha, 2007)
L_∞	$d(u,v) = \max_{i=1}^{n} \lvert u_i - v_i \rvert$	(Cha, 2007)
Lorentzian	$d(u,v) = \sum_{i=1}^{n} ln(1 + \lvert u_i - v_i \rvert)$	(Cha, 2007)
MahalanobisMod	$d(u,v) = \sqrt{\sum_{i=1}^{n} p_i}$ $p_i = \begin{cases} \frac{(u_i - v_i)^2}{\lvert (u_i - \bar{u}) * (v_i - \bar{v}) \rvert}, & (u_i - \bar{u}) * (v_i - \bar{v}) \neq 0 \\ 0, & otherwise \end{cases}$	based on (Deza and Deza, 2016)
Mb	$s(u,v) = \frac{1}{2}\left(\frac{\sum_{i=1}^{n} u_i * v_i}{\sum_{i=1}^{n} u_i^2} + \frac{\sum_{i=1}^{n} u_i * v_i}{\sum_{i=1}^{n} v_i^2}\right)$	(Deza and Deza, 2016)
MBAdjCos	$s(u,v) = \begin{cases} 1, & Mb(u,v) \geq \lambda \\ \frac{Mb(u,v)}{\lambda}, & otherwise \end{cases}$ $\lambda = 0.1$	
MBAdjCosPFMod	$s(u,v) = \begin{cases} 1, & MBPFMod(u,v) \geq \lambda \\ \frac{MBPFMod(u,v)}{\lambda}, & otherwise \end{cases}$ $\lambda = 1$	

	Definition	Reference
MBAdjCosAM	$s(u,v) = \frac{Mb(u,v) + AdjCos(u,v)}{2}$	
MBAdjCosGM	$s(u,v) = \sqrt{Mb(u,v) * AdjCos(u,v)}$	
MBAdjCosHM	$s(u,v) = \frac{2 * Mb(u,v) * AdjCos(u,v)}{Mb(u,v) + AdjCos(u,v)}$	
MBAdjCosProd	$s(u,v) = Mb(u,v) * AdjCos(u,v)$	
MBAdjCosLogProd	$s(u,v) = t_{lb}(Mb(u,v)) * t_{lb}(AdjCos(u,v))$	
MBCosAM	$s(u,v) = \frac{Mb(u,v) + Cos(u,v)}{2}$	
MBCosGM	$s(u,v) = \sqrt{Mb(u,v) * Cos(u,v)}$	
MBCosHM	$s(u,v) = \frac{2 * Mb(u,v) * Cos(u,v)}{Mb(u,v) + Cos(u,v)}$	
MBCosProd	$s(u,v) = Mb(u,v) * Cos(u,v)$	
MBCosLogProd	$s(u,v) = t_{lb}(Mb(u,v)) * t_{lb}(Cos(u,v))$	
MBPFMod	$s(u,v) = \frac{1}{2}\left(\frac{\sum_{i=1}^{n} u_i * v_i}{\sum_{i=1}^{n} u_i^2} + \frac{\sum_{i=1}^{n} u_i * v_i}{\sum_{i=1}^{n} v_i^2}\right) * \frac{1}{2} * \left(lb\left(\frac{N^*}{c(u)}\right) + lb\left(\frac{N^*}{c(v)}\right)\right)$	
Multiplicative	$d(u,v) = -1 + \prod_{i=1}^{n} 1 + \lvert u_i - v_i \rvert$	(Deza and Deza, 2016)
MultiplicativeMod1	$d(u,v) = -1 + \prod_{i=1}^{n} (1 + \lvert u_i - v_i \rvert)^{0.1}$	based on (Deza and Deza, 2016)
MultiplicativeMod2	$d(u,v) = -1 + \prod_{i=1}^{n} lb(1 + \lvert u_i - v_i \rvert)$	based on (Deza and Deza, 2016)
NCDMod1	$d(u,v) = \sum_{i=1}^{n} \frac{u_i * v_i - min(u_i, v_i)}{max(u_i, v_i)}$	inspired by (Cilibrasi and Vitányi, 2007)
NCDMod2	$d(u,v) = \frac{\sum_{i=1}^{n} u_i * v_i - \sum_{i=1}^{n} min(u_i, v_i)}{\sum_{i=1}^{n} max(u_i, v_i)}$	inspired by (Cilibrasi and Vitányi, 2007)
NGDMod	$d(u,v) = \sum_{i=1}^{n} \frac{max(t_{lb}(u_i), t_{lb}(v_i)) - t_{lb}(u_i * v_i)}{t_{lb}(n) - min(t_{lb}(u_i), t_{lb}(v_i))}$	inspired by (Cilibrasi and Vitányi, 2007)

	Definition	Reference
NormCosMod	$s(u,v) = \begin{cases} 1, & CosHM(u,v) \geq \lambda \\ \frac{CosHM(u,v)}{\lambda}, & otherwise \end{cases}$ $CosHM(u,v) = \frac{\sum_{i=1}^{n} sgn(u_i * v_i) * \lvert u_i * v_i \rvert^{\gamma}}{\sqrt{\sum_{i=1}^{n} \lvert u_i \rvert^{2\gamma}} * \sqrt{\sum_{i=1}^{n} \lvert v_i \rvert^{2\gamma}}}$ $\lambda = 0.01 \quad \gamma = 0.05$	(Hassan and Mihalcea, 2011)
NormModSOCPMIMod	$s(u,v) = \begin{cases} 1, & SocHM(u,v) \geq \lambda \\ \frac{SocHM(u,v)}{\lambda}, & otherwise \end{cases}$ $SocHM(u,v) = ln\left(\frac{f(u,v)}{b(u)} + \frac{f(v,u)}{b(v)} + 1\right)$ $f(x,y) = \sum_{i=1}^{b(x)} \begin{cases} sgn(x_i) * \lvert x_i \rvert^{\gamma}, & x_i > 0 \ \wedge y_i > 0 \\ 0, & otherwise \end{cases}$ $b(x) = lg(c(x))^2 * \frac{lb(N)}{\delta} \quad \lambda = 0.125 \quad \gamma = 1 \quad \delta = 0.3$ vectors are sorted and reduced according to the original method	based on (Hassan and Mihalcea, 2011)
Overlap		(Jones and Furnas, 1987)
Pears	$s(u,v) = \frac{\sum_{i=1}^{n}(u_i - \bar{u}) * (v_i - \bar{v})}{\sqrt{\sum_{i=1}^{n}(u_i - \bar{u})^2} * \sqrt{\sum_{i=1}^{n}(v_i - \bar{v})^2}}$	(Jones and Furnas, 1987)
PearsAdjCos	$s(u,v) = \begin{cases} 1, & Pears(u,v) \geq \lambda \\ \frac{Pears(u,v)}{\lambda}, & otherwise \end{cases}$ $\lambda = 0.1$	
PearsAdjCosPFMod	$s(u,v) = \begin{cases} 1, & PearsPFMod(u,v) \geq \lambda \\ \frac{PearsPFMod(u,v)}{\lambda}, & otherwise \end{cases}$ $\lambda = 1$	
PearsMB	$s(u,v) = \frac{1}{2}\left(\frac{\sum_{i=1}^{n}(u_i - \bar{u}) * (v_i - \bar{v})}{\sum_{i=1}^{n}(u_i - \bar{u})^2} + \frac{\sum_{i=1}^{n}(u_i - \bar{u}) * (v_i - \bar{v})}{\sum_{i=1}^{n}(v_i - \bar{v})^2}\right)$	
PearsMBAdjCos	$s(u,v) = \begin{cases} 1, & PearsMB(u,v) \geq \lambda \\ \frac{PearsMB(u,v)}{\lambda}, & otherwise \end{cases}$ $\lambda = 0.1$	
PearsMBAdjCosPFMod	$s(u,v) = \begin{cases} 1, & PearsMBPFMod(u,v) \geq \lambda \\ \frac{PearsMBPFMod(u,v)}{\lambda}, & otherwise \end{cases}$ $\lambda = 1$	
PearsMBPFMod	$s(u,v) = \frac{1}{2}\left(\frac{\sum_{i=1}^{n}(u_i - \bar{u}) * (v_i - \bar{v})}{\sum_{i=1}^{n}(u_i - \bar{u})^2} + \frac{\sum_{i=1}^{n}(u_i - \bar{u}) * (v_i - \bar{v})}{\sum_{i=1}^{n}(v_i - \bar{v})^2}\right) * \frac{1}{2} * \left(lb\left(\frac{N^*}{c(u)}\right) + lb\left(\frac{N^*}{c(v)}\right)\right)$	

	Definition	Reference
PearsPFMod	$s(u,v) = \frac{\sum_{i=1}^{n}(u_i-\bar{u})*(v_i-\bar{v})}{\sqrt{\sum_{i=1}^{n}(u_i-\bar{u})^2}*\sqrt{\sum_{i=1}^{n}(v_i-\bar{v})^2}} * \frac{1}{2}*\left(lb\left(\frac{N^*}{c(u)}\right)+lb\left(\frac{N^*}{c(v)}\right)\right)$	
PenroseShape	$d(u,v) = \sqrt{\sum_{i=1}^{n}((u_i-\bar{u})-(v_i-\bar{v}))^2}$	(Deza and Deza, 2016)
PSChi^{2}Mod		based on (Cha, 2007)
PseudoCos	$s(u,v) = \frac{\sum_{i=1}^{n} u_i * v_i}{\sum_{i=1}^{n} u_i * \sum_{i=1}^{n} v_i}$	(Jones and Furnas, 1987)
PseudoCosMod1	$s(u,v) = \frac{\sum_{i=1}^{n} u_i * v_i}{\sum_{i=1}^{n} \lvert u_i\rvert * \sum_{i=1}^{n} \lvert v_i\rvert}$	based on (Jones and Furnas, 1987)
PseudoCosMod2	$s(u,v) = \begin{cases} \frac{\sum_{i=1}^{n} u_i*v_i}{\sqrt{\sum_{i=1}^{n} u_i}*\sqrt{\sum_{i=1}^{n} v_i}}, & \sum_{i=1}^{n} u_i > 0 \ \wedge \sum_{i=1}^{n} v_i > 0 \\ 0, & otherwise \end{cases}$	based on (Jones and Furnas, 1987)
PseudoCosMod3	$s(u,v) = \frac{\sum_{i=1}^{n} u_i * v_i}{\sqrt{\sum_{i=1}^{n} \lvert u_i\rvert} * \sqrt{\sum_{i=1}^{n} \lvert v_i\rvert}}$	based on (Jones and Furnas, 1987)
RBO	$s(u,v) = (1-p)\sum_{i=1}^{\lvert H\rvert} p^{i-1}\frac{\lvert H_i\rvert}{i}$ $H = \{f \lvert u_f \neq 0 \ \wedge v_f \neq 0\}$ H_d: set of overlapping dimensions between the top d elements of u and v	(Pilehvar and Navigli, 2015)
RényiDivMod_2	$d(u,v) = lb\left(\sum_{i=1}^{n}\begin{cases}\frac{u_i^2}{\lvert v_i\rvert}, & v_i \neq 0 \\ 0, & v_i = 0\end{cases}\right)$	based on (Rényi, 1961)
Rényi$\text{DivMod}_{\text{Inf}}$	$d(u,v) = lb\left(\max_{i=1}^{n}\begin{cases}\left\lvert\frac{u_i}{v_i}\right\rvert, & v_i \neq 0 \\ 0, & v_i = 0\end{cases}\right)$	based on (Rényi, 1961)
RoberstMod	$s(u,v) = \frac{\sum_{i=1}^{n} p_i}{\sum_{i=1}^{n}\lvert u_i\rvert + \sum_{i=1}^{n}\lvert v_i\rvert}$ $p_i = \begin{cases}\frac{(u_i+v_i)*min(u_i,v_i)}{max(u_i,v_i)}, & u_i \neq 0 \ \wedge v_i \neq 0 \\ 0, & otherwise\end{cases}$	basd on (Deza and Deza, 2016)
RMS	$\sum_{i=1}^{n}\sqrt{\frac{u_i^2+v_i^2}{2}}$	(Wonnacott and Wonnacott, 1990)
Simpson1	$s(u,v) = \frac{\sum_{i=1}^{n} u_i * v_i}{min\left(\sum_{i=1}^{n}\lvert u_i\rvert, \sum_{i=1}^{n}\lvert v_i\rvert\right)}$	(Deza and Deza, 2016)
Simpson2Mod	$s(u,v) = \frac{\sum_{i=1}^{n} t_{lb}(u_i * v_i)}{min\left(\sum_{i=1}^{n} lb(\lvert u_i\rvert+1), \sum_{i=1}^{n} lb(\lvert v_i\rvert+1)\right)}$	based on (Jabeen, 2014)
SmoothCos	$s(u,v) = \frac{(\sum_{i=1}^{n} u_i * v_i) + 0.1^2}{\sqrt{(\sum_{i=1}^{n} u_i^2) + 0.1^2} * \sqrt{(\sum_{i=1}^{n} v_i^2) + 0.1^2}}$	(Riedl, 2016)

	Definition	Reference
SOCPMIMod	$s(u,v) = \frac{f(u,v)}{b(u)} + \frac{f(v,u)}{b(v)}$ $f(x,y) = \sum_{i=1}^{b(x)} \begin{cases} sgn(x_i) * \lvert x_i \rvert^{\gamma}, & x_i > 0 \ \wedge y_i > 0 \\ 0, & otherwise \end{cases}$ $b(x) = lg(c(x))^2 * \frac{lb(N)}{\mu} \quad \gamma = 3 \quad \mu = 6.5$ vectors are sorted and reduced according to the original method	based on (Islam and Inkpen, 2008)
SorensenMod	$d(u,v) = \frac{\sum_{i=1}^{n} \lvert u_i - v_i \rvert}{\sum_{i=1}^{n} \lvert u_i \rvert + \sum_{i=1}^{n} \lvert v_i \rvert}$	based on (Deza and Deza, 2016)
Spearman	$s(u,v) = Pears(ranks(u), ranks(v))$	(Deza and Deza, 2016)
Spline	$s(u,v) = \sum_{i=1}^{n} 1 + u_i * v_i + u_i * v_i * min(u_i, v_i) - \frac{u_i + v_i}{2}(min(u_i, v_i))^2 + \frac{(min(u_i, v_i))^3}{3}$	(Vapnik et al., 1997)
StdLike1	$d(u,v) = \sqrt{\frac{\sum_{i=1}^{n} \sqrt{\lvert u_i - mean(u_i, v_i) \rvert} + \sqrt{\lvert v_i - mean(u_i, v_i) \rvert}}{2n}}$	based on (Wonnacott and Wonnacott, 1990)
StdLike2	$d(u,v) = \sqrt{\frac{\sum_{i=1}^{n} (u_i - mean(u_i, v_i))^2 + (v_i - mean(u_i, v_i))^2}{2n}}$	based on (Wonnacott and Wonnacott, 1990)
StdLike2	$d(u,v) = \sqrt{\frac{\sum_{i=1}^{n} \lvert u_i - mean(u_i, v_i) \rvert^3 + \lvert v_i - mean(u_i, v_i) \rvert^3}{2n}}$	based on (Wonnacott and Wonnacott, 1990)
StdLike2	$d(u,v) = \sqrt{\frac{\sum_{i=1}^{n} lb\left(\lvert u_i - mean(u_i, v_i) \rvert + 1\right) + lb\left(\lvert v_i - mean(u_i, v_i) \rvert + 1\right)}{2n}}$	based on (Wonnacott and Wonnacott, 1990)
StdLike2	$d(u,v) = \sqrt{\frac{\sum_{i=1}^{n} \frac{1}{1+e^{-\lvert u_i - mean(u_i, v_i) \rvert}} - 0.5 + \frac{1}{1+e^{-\lvert v_i - mean(u_i, v_i) \rvert}} - 0.5}{2n}}$	based on (Wonnacott and Wonnacott, 1990)
Tanimoto1	$d(u,v) = \frac{\sum_{i=1}^{n} max(u_i, v_i) - min(u_i, v_i)}{\sum_{i=1}^{n} max(u_i, v_i)}$	(Cha, 2007)
Tanimoto1Mod	$d(u,v) = \frac{\sum_{i=1}^{n} max(u_i, v_i) - min(u_i, v_i)}{\sum_{i=1}^{n} \lvert max(u_i, v_i) \rvert}$	based on (Cha, 2007)
VicSymChi^{2}Mod1		based on (Cha, 2007)
VicSymChi^{2}Mod2		based on (Cha, 2007)
VicSymChi^{2}Mod3		based on (Cha, 2007)
WO	$s(u,v) = \frac{\sum_{i=1}^{\lvert H \rvert} (rank(u_i) + rank(v_i))^{-1}}{\sum_{i=1}^{\lvert H \rvert} (2i)^{-1}}$ $H = \{f \mid u_f \neq 0 \ \wedge v_f \neq 0\}$ vectors are sorted according to the original method	(Pilehvar et al., 2013)
ZKLMod	$d(u,v) = \sum_{i=1}^{n} u_i * \begin{cases} ln\left(\frac{u_i}{v_i}\right), & u_i * v_i > 0 \\ \gamma, & u_i * v_i \leq 0 \end{cases}$ $\gamma = 5$	(Hughes and Ramage, 2007)
ZKLModSym	$d(u,v) = \sum_{i=1}^{n} \begin{cases} u_i * ln\left(\frac{u_i}{v_i}\right) + v_i * ln\left(\frac{v_i}{u_i}\right), & u_i * v_i > 0 \\ u_i * \gamma + v_i * \gamma, & u_i * v_i \leq 0 \end{cases}$ $\gamma = 5$	based on (Hughes and Ramage, 2007; Cha, 2007)

	Definition	Reference
CosMod-1.x	$s(u,v) = \frac{\sum_{i=1}^{n} t_x(u_i) * t_x(v_i)}{\sqrt{\sum_{i=1}^{n} u_i^2} * \sqrt{\sum_{i=1}^{n} v_i^2}}$	
CosMod-2.x	$s(u,v) = \frac{\sum_{i=1}^{n} t_x(u_i) * t_x(v_i)}{\sqrt{\sum_{i=1}^{n} (t_x(u_i)^2)} * \sqrt{\sum_{i=1}^{n} (t_x(v_i)^2)}}$	
CosMod-3.x	$s(u,v) = \frac{\sum_{i=1}^{n} t_x(u_i) * t_x(v_i)}{\sqrt{t_x^{-1}(\sum_{i=1}^{n} (t_x(u_i)^2))} * \sqrt{t_x^{-1}(\sum_{i=1}^{n} (t_x(v_i)^2))}}$	
CosMod-4.x	$s(u,v) = \frac{t_x^{-1}(\sum_{i=1}^{n} t_x(u_i) * t_x(v_i))}{\sqrt{\sum_{i=1}^{n} u_i^2} * \sqrt{\sum_{i=1}^{n} v_i^2}}$	
CosMod-5.x	$s(u,v) = \frac{t_x^{-1}(\sum_{i=1}^{n} t_x(u_i) * t_x(v_i))}{\sqrt{\sum_{i=1}^{n} (t_x(u_i)^2)} * \sqrt{\sum_{i=1}^{n} (t_x(v_i)^2)}}$	
CosMod-6.x	$s(u,v) = \frac{t_x^{-1}(\sum_{i=1}^{n} t_x(u_i) * t_x(v_i))}{\sqrt{t_x^{-1}(\sum_{i=1}^{n} (t_x(u_i)^2))} * \sqrt{t_x^{-1}(\sum_{i=1}^{n} (t_x(v_i)^2))}}$	
MbMod-1.x	$s(u,v) = \frac{1}{2}\left(\frac{\sum_{i=1}^{n} t_x(u_i) * t_x(v_i)}{\sum_{i=1}^{n} u_i^2} + \frac{\sum_{i=1}^{n} t_x(u_i) * t_x(v_i)}{\sum_{i=1}^{n} v_i^2}\right)$	
MbMod-2.x	$s(u,v) = \frac{1}{2}\left(\frac{\sum_{i=1}^{n} t_x(u_i) * t_x(v_i)}{\sum_{i=1}^{n} (t_x(u_i)^2)} + \frac{\sum_{i=1}^{n} t_x(u_i) * t_x(v_i)}{\sum_{i=1}^{n} (t_x(v_i)^2)}\right)$	
MbMod-3.x	$s(u,v) = \frac{1}{2}\left(\frac{\sum_{i=1}^{n} t_x(u_i) * t_x(v_i)}{t_x^{-1}(\sum_{i=1}^{n} (t_x(u_i)^2))} + \frac{\sum_{i=1}^{n} t_x(u_i) * t_x(v_i)}{t_x^{-1}(\sum_{i=1}^{n} (t_x(v_i)^2))}\right)$	
MbMod-4.x	$s(u,v) = \frac{1}{2}\left(\frac{t_x^{-1}(\sum_{i=1}^{n} t_x(u_i) * t_x(v_i))}{\sum_{i=1}^{n} u_i^2} + \frac{t_x^{-1}(\sum_{i=1}^{n} t_x(u_i) * t_x(v_i))}{\sum_{i=1}^{n} v_i^2}\right)$	
MbMod-5.x	$s(u,v) = \frac{1}{2}\left(\frac{t_x^{-1}(\sum_{i=1}^{n} t_x(u_i) * t_x(v_i))}{\sum_{i=1}^{n} (t_x(u_i)^2)} + \frac{t_x^{-1}(\sum_{i=1}^{n} t_x(u_i) * t_x(v_i))}{\sum_{i=1}^{n} (t_x(v_i)^2)}\right)$	
MbMod-6.x	$s(u,v) = \frac{1}{2}\left(\frac{t_x^{-1}(\sum_{i=1}^{n} t_x(u_i) * t_x(v_i))}{t_x^{-1}(\sum_{i=1}^{n} (t_x(u_i)^2))} + \frac{t_x^{-1}(\sum_{i=1}^{n} t_x(u_i) * t_x(v_i))}{t_x^{-1}(\sum_{i=1}^{n} (t_x(v_i)^2))}\right)$	

	Definition	Reference
AdjCosMod-1.x	$s(u,v) = \begin{cases} 1, & CosMod\text{-}1.x(u,v) \geq \lambda \\ \frac{CosMod\text{-}1.x(u,v)}{\lambda}, & otherwise \end{cases}$ $\lambda = 0.1$	
AdjCosMod-2.x	$s(u,v) = \begin{cases} 1, & CosMod\text{-}2.x(u,v) \geq \lambda \\ \frac{CosMod\text{-}2.x(u,v)}{\lambda}, & otherwise \end{cases}$ $\lambda = 0.1$	
AdjCosMod-3.x	$s(u,v) = \begin{cases} 1, & CosMod\text{-}3.x(u,v) \geq \lambda \\ \frac{CosMod\text{-}3.x(u,v)}{\lambda}, & otherwise \end{cases}$ $\lambda = 0.1$	
AdjCosMod-4.x	$s(u,v) = \begin{cases} 1, & CosMod\text{-}4.x(u,v) \geq \lambda \\ \frac{CosMod\text{-}4.x(u,v)}{\lambda}, & otherwise \end{cases}$ $\lambda = 0.1$	
AdjCosMod-5.x	$s(u,v) = \begin{cases} 1, & CosMod\text{-}5.x(u,v) \geq \lambda \\ \frac{CosMod\text{-}5.x(u,v)}{\lambda}, & otherwise \end{cases}$ $\lambda = 0.1$	
AdjCosMod-6.x	$s(u,v) = \begin{cases} 1, & CosMod\text{-}6.x(u,v) \geq \lambda \\ \frac{CosMod\text{-}6.x(u,v)}{\lambda}, & otherwise \end{cases}$ $\lambda = 0.1$	
PFMod-1.x	$s(u,v) = CosMod\text{-}1.x * \left(lb\left(\frac{N^*}{c(u)}\right) + lb\left(\frac{N^*}{c(v)}\right)\right)$	
PFMod-2.x	$s(u,v) = CosMod\text{-}2.x * \left(lb\left(\frac{N^*}{c(u)}\right) + lb\left(\frac{N^*}{c(v)}\right)\right)$	
PFMod-3.x	$s(u,v) = CosMod\text{-}3.x * \left(lb\left(\frac{N^*}{c(u)}\right) + lb\left(\frac{N^*}{c(v)}\right)\right)$	
PFMod-4.x	$s(u,v) = CosMod\text{-}4.x * \left(lb\left(\frac{N^*}{c(u)}\right) + lb\left(\frac{N^*}{c(v)}\right)\right)$	
PFMod-5.x	$s(u,v) = CosMod\text{-}5.x * \left(lb\left(\frac{N^*}{c(u)}\right) + lb\left(\frac{N^*}{c(v)}\right)\right)$	
PFMod-6.x	$s(u,v) = CosMod\text{-}6.x * \left(lb\left(\frac{N^*}{c(u)}\right) + lb\left(\frac{N^*}{c(v)}\right)\right)$	
InnerProdW-1.x	$s(u,v) = \sum_{i=1}^{n} \frac{t_x(u_i) * t_x(v_i)}{\lvert u_i\rvert + \lvert v_i\rvert}$	
InnerProdW-2.x	$s(u,v) = \sum_{i=1}^{n} \frac{t_x(u_i) * t_x(v_i)}{\lvert t_x(u_i)\rvert + \lvert t_x(v_i)\rvert}$	
InnerProdW-3.x	$s(u,v) = t_x^{-1}\left(\sum_{i=1}^{n} \frac{t_x(u_i) * t_x(v_i)}{\lvert u_i\rvert + \lvert v_i\rvert}\right)$	
InnerProdW-4.x	$s(u,v) = t_x^{-1}\left(\sum_{i=1}^{n} \frac{t_x(u_i) * t_x(v_i)}{\lvert t_x(u_i)\rvert + \lvert t_x(v_i)\rvert}\right)$	

	Definition	Reference
PearsMod-1.x	$s(u,v) = \frac{\sum_{i=1}^{n} t_x(u_i - \bar{u}) * t_x(v_i - \bar{v})}{\sqrt{\sum_{i=1}^{n}(u_i - \bar{u})^2} * \sqrt{\sum_{i=1}^{n}(v_i - \bar{v})^2}}$	
PearsMod-2.x	$s(u,v) = \frac{\sum_{i=1}^{n} t_x(u_i - \bar{u}) * t_x(v_i - \bar{v})}{\sqrt{\sum_{i=1}^{n}(t_x(u_i - \bar{u}))^2} * \sqrt{\sum_{i=1}^{n}(t_x(v_i - \bar{v}))^2}}$	
PearsMod-3.x	$s(u,v) = \frac{\sum_{i=1}^{n} t_x(u_i - \bar{u}) * t_x(v_i - \bar{v})}{\sqrt{t_x^{-1}\left(\sum_{i=1}^{n}(t_x(u_i - \bar{u}))^2\right)} * \sqrt{t_x^{-1}\left(\sum_{i=1}^{n}(t_x(v_i - \bar{v}))^2\right)}}$	
PearsMod-4.x	$s(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i - \bar{u}) * t_x(v_i - \bar{v})\right)}{\sqrt{\sum_{i=1}^{n}(u_i - \bar{u})^2} * \sqrt{\sum_{i=1}^{n}(v_i - \bar{v})^2}}$	
PearsMod-5.x	$s(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i - \bar{u}) * t_x(v_i - \bar{v})\right)}{\sqrt{\sum_{i=1}^{n}(t_x(u_i - \bar{u}))^2} * \sqrt{\sum_{i=1}^{n}(t_x(v_i - \bar{v}))^2}}$	
PearsMod-6.x	$s(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i - \bar{u}) * t_x(v_i - \bar{v})\right)}{\sqrt{t_x^{-1}\left(\sum_{i=1}^{n}(t_x(u_i - \bar{u}))^2\right)} * \sqrt{t_x^{-1}\left(\sum_{i=1}^{n}(t_x(v_i - \bar{v}))^2\right)}}$	
PearsMBMod-1.x	$s(u,v) = \frac{1}{2}\left(\frac{\sum_{i=1}^{n} t_x(u_i - \bar{u}) * t_x(v_i - \bar{v})}{\sum_{i=1}^{n}(u_i - \bar{u})^2} + \frac{\sum_{i=1}^{n} t_x(u_i - \bar{u}) * t_x(v_i - \bar{v})}{\sum_{i=1}^{n}(v_i - \bar{v})^2}\right)$	
PearsMBMod-2.x	$s(u,v) = \frac{1}{2}\left(\frac{\sum_{i=1}^{n} t_x(u_i - \bar{u}) * t_x(v_i - \bar{v})}{\sum_{i=1}^{n}(t_x(u_i - \bar{u}))^2} + \frac{\sum_{i=1}^{n} t_x(u_i - \bar{u}) * t_x(v_i - \bar{v})}{\sum_{i=1}^{n}(t_x(v_i - \bar{v}))^2}\right)$	
PearsMBMod-3.x	$s(u,v) = \frac{1}{2}\left(\frac{\sum_{i=1}^{n} t_x(u_i - \bar{u}) * t_x(v_i - \bar{v})}{t_x^{-1}\left(\sum_{i=1}^{n}(t_x(u_i - \bar{u}))^2\right)} + \frac{\sum_{i=1}^{n} t_x(u_i - \bar{u}) * t_x(v_i - \bar{v})}{t_x^{-1}\left(\sum_{i=1}^{n}(t_x(v_i - \bar{v}))^2\right)}\right)$	
PearsMBMod-4.x	$s(u,v) = \frac{1}{2}\left(\frac{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i - \bar{u}) * t_x(v_i - \bar{v})\right)}{\sum_{i=1}^{n}(u_i - \bar{u})^2} + \frac{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i - \bar{u}) * t_x(v_i - \bar{v})\right)}{\sum_{i=1}^{n}(v_i - \bar{v})^2}\right)$	
PearsMBMod-5.x	$s(u,v) = \frac{1}{2}\left(\frac{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i - \bar{u}) * t_x(v_i - \bar{v})\right)}{\sum_{i=1}^{n}(t_x(u_i - \bar{u}))^2} + \frac{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i - \bar{u}) * t_x(v_i - \bar{v})\right)}{\sum_{i=1}^{n}(t_x(v_i - \bar{v}))^2}\right)$	
PearsMBMod-6.x	$s(u,v) = \frac{1}{2}\left(\frac{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i - \bar{u}) * t_x(v_i - \bar{v})\right)}{t_x^{-1}\left(\sum_{i=1}^{n}(t_x(u_i - \bar{u}))^2\right)} + \frac{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i - \bar{u}) * t_x(v_i - \bar{v})\right)}{t_x^{-1}\left(\sum_{i=1}^{n}(t_x(v_i - \bar{v}))^2\right)}\right)$	

	Definition	Reference
PearsMBAdjCosMod-1.x	$s(u,v) = \begin{cases} 1, & PearsMBMod\text{-}1.x(u,v) \geq \lambda \\ \frac{PearsMBMod\text{-}1.x(u,v)}{\lambda}, & otherwise \end{cases}$ $\lambda = 0.1$	
PearsMBAdjCosMod-2.x	$s(u,v) = \begin{cases} 1, & PearsMBMod\text{-}2.x(u,v) \geq \lambda \\ \frac{PearsMBMod\text{-}2.x(u,v)}{\lambda}, & otherwise \end{cases}$ $\lambda = 0.1$	
PearsMBAdjCosMod-3.x	$s(u,v) = \begin{cases} 1, & PearsMBMod\text{-}3.x(u,v) \geq \lambda \\ \frac{PearsMBMod\text{-}3.x(u,v)}{\lambda}, & otherwise \end{cases}$ $\lambda = 0.1$	
PearsMBAdjCosMod-4.x	$s(u,v) = \begin{cases} 1, & PearsMBMod\text{-}4.x(u,v) \geq \lambda \\ \frac{PearsMBMod\text{-}4.x(u,v)}{\lambda}, & otherwise \end{cases}$ $\lambda = 0.1$	
PearsMBAdjCosMod-5.x	$s(u,v) = \begin{cases} 1, & PearsMBMod\text{-}5.x(u,v) \geq \lambda \\ \frac{PearsMBMod\text{-}5.x(u,v)}{\lambda}, & otherwise \end{cases}$ $\lambda = 0.1$	
PearsMBAdjCosMod-6.x	$s(u,v) = \begin{cases} 1, & PearsMBMod\text{-}6.x(u,v) \geq \lambda \\ \frac{PearsMBMod\text{-}6.x(u,v)}{\lambda}, & otherwise \end{cases}$ $\lambda = 0.1$	
PearsMBAdjCosPFMod-1.x	$s(u,v) = \begin{cases} 1, & PearsMBMod\text{-}1.x(u,v) * \left(lb\left(\frac{N^*}{c(u)}\right) + lb\left(\frac{N^*}{c(v)}\right)\right) \geq \lambda \\ \frac{PearsMBMod\text{-}1.x(u,v) * \left(lb\left(\frac{N^*}{c(u)}\right) + lb\left(\frac{N^*}{c(v)}\right)\right)}{\lambda}, & otherwise \end{cases}$ $\lambda = 0.1$	
PearsMBAdjCosPFMod-2.x	$s(u,v) = \begin{cases} 1, & PearsMBMod\text{-}2.x(u,v) * \left(lb\left(\frac{N^*}{c(u)}\right) + lb\left(\frac{N^*}{c(v)}\right)\right) \geq \lambda \\ \frac{PearsMBMod\text{-}2.x(u,v) * \left(lb\left(\frac{N^*}{c(u)}\right) + lb\left(\frac{N^*}{c(v)}\right)\right)}{\lambda}, & otherwise \end{cases}$ $\lambda = 0.1$	
PearsMBAdjCosPFMod-3.x	$s(u,v) = \begin{cases} 1, & PearsMBMod\text{-}3.x(u,v) * \left(lb\left(\frac{N^*}{c(u)}\right) + lb\left(\frac{N^*}{c(v)}\right)\right) \geq \lambda \\ \frac{PearsMBMod\text{-}3.x(u,v) * \left(lb\left(\frac{N^*}{c(u)}\right) + lb\left(\frac{N^*}{c(v)}\right)\right)}{\lambda}, & otherwise \end{cases}$ $\lambda = 0.1$	
PearsMBAdjCosPFMod-4.x	$s(u,v) = \begin{cases} 1, & PearsMBMod\text{-}4.x(u,v) * \left(lb\left(\frac{N^*}{c(u)}\right) + lb\left(\frac{N^*}{c(v)}\right)\right) \geq \lambda \\ \frac{PearsMBMod\text{-}4.x(u,v) * \left(lb\left(\frac{N^*}{c(u)}\right) + lb\left(\frac{N^*}{c(v)}\right)\right)}{\lambda}, & otherwise \end{cases}$ $\lambda = 0.1$	
PearsMBAdjCosPFMod-5.x	$s(u,v) = \begin{cases} 1, & PearsMBMod\text{-}5.x(u,v) * \left(lb\left(\frac{N^*}{c(u)}\right) + lb\left(\frac{N^*}{c(v)}\right)\right) \geq \lambda \\ \frac{PearsMBMod\text{-}5.x(u,v) * \left(lb\left(\frac{N^*}{c(u)}\right) + lb\left(\frac{N^*}{c(v)}\right)\right)}{\lambda}, & otherwise \end{cases}$ $\lambda = 0.1$	
PearsMBAdjCosPFMod-6.x	$s(u,v) = \begin{cases} 1, & PearsMBMod\text{-}6.x(u,v) * \left(lb\left(\frac{N^*}{c(u)}\right) + lb\left(\frac{N^*}{c(v)}\right)\right) \geq \lambda \\ \frac{PearsMBMod\text{-}6.x(u,v) * \left(lb\left(\frac{N^*}{c(u)}\right) + lb\left(\frac{N^*}{c(v)}\right)\right)}{\lambda}, & otherwise \end{cases}$ $\lambda = 0.1$	

	Definition	Reference
Cos	$s(u,v) = \frac{\sum_{i=1}^{n} u_i * v_i}{\sqrt{\sum_{i=1}^{n} u_i^2} * \sqrt{\sum_{i=1}^{n} v_i^2}}$	(Jones and Furnas, 1987)
LW-1.1.x	$d(u,v) = \sum_{i=1}^{n} \frac{t_x(\lvert u_i - v_i \rvert)}{\lvert u_i \rvert + \lvert v_i \rvert}$	
LW-1.2.x	$d(u,v) = t_x^{-1}\left(\sum_{i=1}^{n} \frac{t_x(\lvert u_i - v_i \rvert)}{\lvert u_i \rvert + \lvert v_i \rvert}\right)$	
LW-2.1.x	$d(u,v) = \sum_{i=1}^{n} \frac{t_x(\lvert u_i - v_i \rvert)}{\lvert t_x(u_i) \rvert + \lvert t_x(v_i) \rvert}$	
LW-2.2.x	$d(u,v) = t_x^{-1}\left(\sum_{i=1}^{n} \frac{t_x(\lvert u_i - v_i \rvert)}{\lvert t_x(u_i) \rvert + \lvert t_x(v_i) \rvert}\right)$	
DTVW-1.1.x	$d(u,v) = \sum_{i=1}^{n} \frac{\lvert t_x(u_i) - t_x(v_i) \rvert}{\lvert u_i \rvert + \lvert v_i \rvert}$	
DTVW-1.2.x	$d(u,v) = t_x^{-1}\left(\sum_{i=1}^{n} \frac{\lvert t_x(u_i) - t_x(v_i) \rvert}{\lvert u_i \rvert + \lvert v_i \rvert}\right)$	
DTVW-2.1.x	$d(u,v) = \sum_{i=1}^{n} \frac{\lvert t_x(u_i) - t_x(v_i) \rvert}{\lvert t_x(u_i) \rvert + \lvert t_x(v_i) \rvert}$	
DTVW-2.2.x	$d(u,v) = t_x^{-1}\left(\sum_{i=1}^{n} \frac{\lvert t_x(u_i) - t_x(v_i) \rvert}{\lvert t_x(u_i) \rvert + \lvert t_x(v_i) \rvert}\right)$	
DTVW-3.1.x	$d(u,v) = \sum_{i=1}^{n} \frac{t_x^{-1}(\lvert t_x(u_i) - t_x(v_i) \rvert)}{\lvert u_i \rvert + \lvert v_i \rvert}$	
DTVW-3.2.x	$d(u,v) = t_x^{-1}\left(\sum_{i=1}^{n} \frac{t_x^{-1}(\lvert t_x(u_i) - t_x(v_i) \rvert)}{\lvert u_i \rvert + \lvert v_i \rvert}\right)$	
DTVW-4.1.x	$d(u,v) = \sum_{i=1}^{n} \frac{t_x^{-1}(\lvert t_x(u_i) - t_x(v_i) \rvert)}{\lvert t_x(u_i) \rvert + \lvert t_x(v_i) \rvert}$	
DTVW-4.2.x	$d(u,v) = t_x^{-1}\left(\sum_{i=1}^{n} \frac{t_x^{-1}(\lvert t_x(u_i) - t_x(v_i) \rvert)}{\lvert t_x(u_i) \rvert + \lvert t_x(v_i) \rvert}\right)$	
PenroseShapeMod-1.x	$d(u,v) = \sum_{i=1}^{n} t_x(\lvert (u_i - \bar{u}) - (v_i - \bar{v}) \rvert)$	
PenroseShapeMod-2.x	$d(u,v) = \sum_{i=1}^{n} \lvert t_x(u_i - \bar{u}) - t_x(v_i - \bar{v}) \rvert$	
PenroseShapeMod-3.x	$d(u,v) = t_x^{-1}\left(\sum_{i=1}^{n} t_x(\lvert (u_i - \bar{u}) - (v_i - \bar{v}) \rvert)\right)$	
PenroseShapeMod-4.x	$d(u,v) = t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(u_i - \bar{u}) - t_x(v_i - \bar{v}) \rvert\right)$	

	Definition	Reference
LMod-1.1.x	$d(u,v) = \sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)$	
LMod-1.2.x	$d(u,v) = t_x^{-1}\left(\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)\right)$	
LMod-2.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)}{\sqrt{\sum_{i=1}^{n} u_i^2} * \sqrt{\sum_{i=1}^{n} v_i^2}}$	
LMod-2.2.x	$d(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)\right)}{\sqrt{\sum_{i=1}^{n} u_i^2} * \sqrt{\sum_{i=1}^{n} v_i^2}}$	
LMod-3.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)}{\sqrt{\sum_{i=1}^{n} u_i^2} + \sqrt{\sum_{i=1}^{n} v_i^2}}$	
LMod-3.2.x	$d(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)\right)}{\sqrt{\sum_{i=1}^{n} u_i^2} + \sqrt{\sum_{i=1}^{n} v_i^2}}$	
LMod-4.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)}{\sqrt{\sum_{i=1}^{n} t_x(u_i)^2} * \sqrt{\sum_{i=1}^{n} t_x(v_i)^2}}$	
LMod-4.2.x	$d(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)\right)}{\sqrt{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i)^2\right)} * \sqrt{t_x^{-1}\left(\sum_{i=1}^{n} t_x(v_i)^2\right)}}$	
LMod-5.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)}{\sqrt{\sum_{i=1}^{n} t_x(u_i)^2} + \sqrt{\sum_{i=1}^{n} t_x(v_i)^2}}$	
LMod-5.2.x	$d(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)\right)}{\sqrt{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i)^2\right)} + \sqrt{t_x^{-1}\left(\sum_{i=1}^{n} t_x(v_i)^2\right)}}$	
LMod-6.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)}{\sqrt{\sum_{i=1}^{n} \lvert t_x(u_i) \rvert} * \sqrt{\sum_{i=1}^{n} \lvert t_x(v_i) \rvert}}$	
LMod-6.2.x	$d(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)\right)}{\sqrt{t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(u_i) \rvert\right)} * \sqrt{t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(v_i) \rvert\right)}}$	
LMod-7.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)}{\sqrt{\sum_{i=1}^{n} \lvert t_x(u_i) \rvert} + \sqrt{\sum_{i=1}^{n} \lvert t_x(v_i) \rvert}}$	
LMod-7.2.x	$d(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)\right)}{\sqrt{t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(u_i) \rvert\right)} + \sqrt{t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(v_i) \rvert\right)}}$	
LMod-8.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)}{\sum_{i=1}^{n} u_i^2 * \sum_{i=1}^{n} v_i^2}$	
LMod-8.2.x	$d(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)\right)}{\sum_{i=1}^{n} u_i^2 * \sum_{i=1}^{n} v_i^2}$	
LMod-9.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)}{\sum_{i=1}^{n} u_i^2 + \sum_{i=1}^{n} v_i^2}$	
LMod-9.2.x	$d(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)\right)}{\sum_{i=1}^{n} u_i^2 + \sum_{i=1}^{n} v_i^2}$	
LMod-10.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)}{\sum_{i=1}^{n} t_x(u_i)^2 * \sum_{i=1}^{n} t_x(v_i)^2}$	
LMod-10.2.x	$d(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)\right)}{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i)^2\right) * t_x^{-1}\left(\sum_{i=1}^{n} t_x(v_i)^2\right)}$	

	Definition	Reference
LMod-11.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)}{\sum_{i=1}^{n} t_x(u_i)^2 + \sum_{i=1}^{n} t_x(v_i)^2}$	
LMod-11.2.x	$d(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)\right)}{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i)^2\right) + t_x^{-1}\left(\sum_{i=1}^{n} t_x(v_i)^2\right)}$	
LMod-12.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)}{\sum_{i=1}^{n} \lvert t_x(u_i) \rvert * \sum_{i=1}^{n} \lvert t_x(v_i) \rvert}$	
LMod-12.2.x	$d(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)\right)}{t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(u_i) \rvert\right) * t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(v_i) \rvert\right)}$	
LMod-13.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)}{\sum_{i=1}^{n} \lvert t_x(u_i) \rvert + \sum_{i=1}^{n} \lvert t_x(v_i) \rvert}$	
LMod-13.2.x	$d(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)\right)}{t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(u_i) \rvert\right) + t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(v_i) \rvert\right)}$	
LMod-14.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)}{\sum_{i=1}^{n} t_x(u_i) * \sum_{i=1}^{n} t_x(v_i)}$	
LMod-14.2.x	$d(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)\right)}{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i)\right) * t_x^{-1}\left(\sum_{i=1}^{n} t_x(v_i)\right)}$	
LMod-15.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)}{\sum_{i=1}^{n} t_x(u_i) + \sum_{i=1}^{n} t_x(v_i)}$	
LMod-15.2.x	$d(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} t_x(\lvert u_i - v_i \rvert)\right)}{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i)\right) + t_x^{-1}\left(\sum_{i=1}^{n} t_x(v_i)\right)}$	
DTV-1.1.x	$d(u,v) = \sum_{i=1}^{n} \lvert t_x(u_i) - t_x(v_i) \rvert$	
DTV-1.2.x	$d(u,v) = t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(u_i) - t_x(v_i) \rvert\right)$	
DTV-2.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} \lvert t_x(u_i) - t_x(v_i) \rvert}{\sum_{i=1}^{n} \lvert t_x(u_i) \rvert + \sum_{i=1}^{n} \lvert t_x(v_i) \rvert}$	
DTV-2.2.x	$d(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(u_i) - t_x(v_i) \rvert\right)}{t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(u_i) \rvert\right) + t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(v_i) \rvert\right)}$	
DTV-3.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} \lvert t_x(u_i) - t_x(v_i) \rvert}{\sum_{i=1}^{n} \lvert t_x(u_i) \rvert * \sum_{i=1}^{n} \lvert t_x(v_i) \rvert}$	
DTV-3.2.x	$d(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(u_i) - t_x(v_i) \rvert\right)}{t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(u_i) \rvert\right) * t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(v_i) \rvert\right)}$	
DTV-4.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} \lvert t_x(u_i) - t_x(v_i) \rvert}{\sqrt{\sum_{i=1}^{n} t_x(u_i)^2} + \sqrt{\sum_{i=1}^{n} t_x(v_i)^2}}$	
DTV-4.2.x	$d(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(u_i) - t_x(v_i) \rvert\right)}{\sqrt{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i)^2\right)} + \sqrt{t_x^{-1}\left(\sum_{i=1}^{n} t_x(v_i)^2\right)}}$	
DTV-5.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} \lvert t_x(u_i) - t_x(v_i) \rvert}{\sqrt{\sum_{i=1}^{n} t_x(u_i)^2} * \sqrt{\sum_{i=1}^{n} t_x(v_i)^2}}$	
DTV-5.2.x	$d(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(u_i) - t_x(v_i) \rvert\right)}{\sqrt{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i)^2\right)} * \sqrt{t_x^{-1}\left(\sum_{i=1}^{n} t_x(v_i)^2\right)}}$	
DTV-6.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} \lvert t_x(u_i) - t_x(v_i) \rvert}{\sqrt{\sum_{i=1}^{n} u_i^2} + \sqrt{\sum_{i=1}^{n} v_i^2}}$	
DTV-6.2.x	$d(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(u_i) - t_x(v_i) \rvert\right)}{\sqrt{\sum_{i=1}^{n} u_i^2} + \sqrt{\sum_{i=1}^{n} v_i^2}}$	

	Definition	Reference
DTV-7.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} \lvert t_x(u_i) - t_x(v_i)\rvert}{\sqrt{\sum_{i=1}^{n} u_i^2} * \sqrt{\sum_{i=1}^{n} v_i^2}}$	
DTV-7.2.x	$d(u,v) = \frac{t_x^{-1}(\sum_{i=1}^{n} \lvert t_x(u_i) - t_x(v_i)\rvert)}{\sqrt{\sum_{i=1}^{n} u_i^2} * \sqrt{\sum_{i=1}^{n} v_i^2}}$	
DTV-8.1.x	$d(u,v) = \sum_{i=1}^{n} t_x^{-1}(\lvert t_x(u_i) - t_x(v_i)\rvert)$	
DTV-9.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} t_x^{-1}(\lvert t_x(u_i) - t_x(v_i)\rvert)}{\sum_{i=1}^{n} \lvert t_x(u_i)\rvert + \sum_{i=1}^{n} \lvert t_x(v_i)\rvert}$	
DTV-9.2.x	$d(u,v) = \frac{\sum_{i=1}^{n} t_x^{-1}(\lvert t_x(u_i) - t_x(v_i)\rvert))}{t_x^{-1}(\sum_{i=1}^{n} \lvert t_x(u_i)\rvert) + t_x^{-1}(\sum_{i=1}^{n} \lvert t_x(v_i)\rvert)}$	
DTV-10.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} t_x^{-1}(\lvert t_x(u_i) - t_x(v_i)\rvert)}{\sum_{i=1}^{n} \lvert t_x(u_i)\rvert * \sum_{i=1}^{n} \lvert t_x(v_i)\rvert}$	
DTV-10.2.x	$d(u,v) = \frac{\sum_{i=1}^{n} t_x^{-1}(\lvert t_x(u_i) - t_x(v_i)\rvert))}{t_x^{-1}(\sum_{i=1}^{n} \lvert t_x(u_i)\rvert) * t_x^{-1}(\sum_{i=1}^{n} \lvert t_x(v_i)\rvert)}$	
DTV-11.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} t_x^{-1}(\lvert t_x(u_i) - t_x(v_i)\rvert)}{\sqrt{\sum_{i=1}^{n} t_x(u_i)^2} + \sqrt{\sum_{i=1}^{n} t_x(v_i)^2}}$	
DTV-11.2.x	$d(u,v) = \frac{\sum_{i=1}^{n} t_x^{-1}(\lvert t_x(u_i) - t_x(v_i)\rvert))}{\sqrt{t_x^{-1}(\sum_{i=1}^{n} t_x(u_i)^2) + \sqrt{t_x^{-1}(\sum_{i=1}^{n} t_x(v_i)^2)}}}$	
DTV-12.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} t_x^{-1}(\lvert t_x(u_i) - t_x(v_i)\rvert)}{\sqrt{\sum_{i=1}^{n} t_x(u_i)^2} * \sqrt{\sum_{i=1}^{n} t_x(v_i)^2}}$	
DTV-12.2.x	$d(u,v) = \frac{\sum_{i=1}^{n} t_x^{-1}(\lvert t_x(u_i) - t_x(v_i)\rvert))}{\sqrt{t_x^{-1}(\sum_{i=1}^{n} t_x(u_i)^2)} * \sqrt{t_x^{-1}(\sum_{i=1}^{n} t_x(v_i)^2)}}$	
DTV-13.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} t_x^{-1}(\lvert t_x(u_i) - t_x(v_i)\rvert)}{\sqrt{\sum_{i=1}^{n} u_i^2} + \sqrt{\sum_{i=1}^{n} v_i^2}}$	
DTV-14.1.x	$d(u,v) = \frac{\sum_{i=1}^{n} t_x^{-1}(\lvert t_x(u_i) - t_x(v_i)\rvert)}{\sqrt{\sum_{i=1}^{n} u_i^2} * \sqrt{\sum_{i=1}^{n} v_i^2}}$	
LinMod-1.1.1.x	$s(u,v) = \sum_{i=1}^{n} lanz_{t_x}(u_i, v_i)$	
LinMod-1.1.2.x	$s(u,v) = t_x^{-1}\left(\sum_{i=1}^{n} lanz_{t_x}(u_i, v_i)\right)$	
LinMod-1.2.1.x	$s(u,v) = \sum_{i=1}^{n} lagz_{t_x}(u_i, v_i)$	
LinMod-1.2.2.x	$s(u,v) = t_x^{-1}\left(\sum_{i=1}^{n} lagz_{t_x}(u_i, v_i)\right)$	
LinMod-2.1.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lanz_{t_x}(u_i, v_i)}{\sum_{i=1}^{n} \lvert t_x(u_i)\rvert + \sum_{i=1}^{n} \lvert t_x(v_i)\rvert}$	
LinMod-2.1.2.x	$s(u,v) = \frac{t_x^{-1}(\sum_{i=1}^{n} lanz_{t_x}(u_i, v_i))}{t_x^{-1}(\sum_{i=1}^{n} \lvert t_x(u_i)\rvert) + t_x^{-1}(\sum_{i=1}^{n} \lvert t_x(v_i)\rvert)}$	
LinMod-2.2.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lagz_{t_x}(u_i, v_i)}{\sum_{i=1}^{n} \lvert t_x(u_i)\rvert + \sum_{i=1}^{n} \lvert t_x(v_i)\rvert}$	
LinMod-2.2.2.x	$s(u,v) = \frac{t_x^{-1}(\sum_{i=1}^{n} lagz_{t_x}(u_i, v_i))}{t_x^{-1}(\sum_{i=1}^{n} \lvert t_x(u_i)\rvert) + t_x^{-1}(\sum_{i=1}^{n} \lvert t_x(v_i)\rvert)}$	
LinMod-3.1.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lanz_{t_x}(u_i, v_i)}{\sum_{i=1}^{n} \lvert t_x(u_i)\rvert * \sum_{i=1}^{n} \lvert t_x(v_i)\rvert}$	
LinMod-3.1.2.x	$s(u,v) = \frac{t_x^{-1}(\sum_{i=1}^{n} lanz_{t_x}(u_i, v_i))}{t_x^{-1}(\sum_{i=1}^{n} \lvert t_x(u_i)\rvert) * t_x^{-1}(\sum_{i=1}^{n} \lvert t_x(v_i)\rvert)}$	
LinMod-3.2.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lagz_{t_x}(u_i, v_i)}{\sum_{i=1}^{n} \lvert t_x(u_i)\rvert * \sum_{i=1}^{n} \lvert t_x(v_i)\rvert}$	
LinMod-3.2.2.x	$s(u,v) = \frac{t_x^{-1}(\sum_{i=1}^{n} lagz_{t_x}(u_i, v_i))}{t_x^{-1}(\sum_{i=1}^{n} \lvert t_x(u_i)\rvert) * t_x^{-1}(\sum_{i=1}^{n} \lvert t_x(v_i)\rvert)}$	

	Definition	Reference
LinMod-4.1.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lanz_{t_x}(u_i, v_i)}{\sqrt{\sum_{i=1}^{n} t_x(u_i)^2} + \sqrt{\sum_{i=1}^{n} t_x(v_i)^2}}$	
LinMod-4.1.2.x	$s(u,v) = \frac{t_x^{-1}(\sum_{i=1}^{n} lanz_{t_x}(u_i, v_i))}{\sqrt{t_x^{-1}(\sum_{i=1}^{n} t_x(u_i)^2)} + \sqrt{t_x^{-1}(\sum_{i=1}^{n} t_x(v_i)^2)}}$	
LinMod-4.2.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lagz_{t_x}(u_i, v_i)}{\sqrt{\sum_{i=1}^{n} t_x(u_i)^2} + \sqrt{\sum_{i=1}^{n} t_x(v_i)^2}}$	
LinMod-4.2.2.x	$s(u,v) = \frac{t_x^{-1}(\sum_{i=1}^{n} lagz_{t_x}(u_i, v_i))}{\sqrt{t_x^{-1}(\sum_{i=1}^{n} t_x(u_i)^2)} + \sqrt{t_x^{-1}(\sum_{i=1}^{n} t_x(v_i)^2)}}$	
LinMod-5.1.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lanz_{t_x}(u_i, v_i)}{\sqrt{\sum_{i=1}^{n} t_x(u_i)^2} * \sqrt{\sum_{i=1}^{n} t_x(v_i)^2}}$	
LinMod-5.1.2.x	$s(u,v) = \frac{t_x^{-1}(\sum_{i=1}^{n} lanz_{t_x}(u_i, v_i))}{\sqrt{t_x^{-1}(\sum_{i=1}^{n} t_x(u_i)^2)} * \sqrt{t_x^{-1}(\sum_{i=1}^{n} t_x(v_i)^2)}}$	
LinMod-5.2.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lagz_{t_x}(u_i, v_i)}{\sqrt{\sum_{i=1}^{n} t_x(u_i)^2} * \sqrt{\sum_{i=1}^{n} t_x(v_i)^2}}$	
LinMod-5.2.2.x	$s(u,v) = \frac{t_x^{-1}(\sum_{i=1}^{n} lagz_{t_x}(u_i, v_i))}{\sqrt{t_x^{-1}(\sum_{i=1}^{n} t_x(u_i)^2)} * \sqrt{t_x^{-1}(\sum_{i=1}^{n} t_x(v_i)^2)}}$	
LinMod-6.1.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lanz_{t_x}(u_i, v_i)}{\sqrt{\sum_{i=1}^{n} u_i^2} + \sqrt{\sum_{i=1}^{n} v_i^2}}$	
LinMod-6.1.2.x	$s(u,v) = \frac{t_x^{-1}(\sum_{i=1}^{n} lanz_{t_x}(u_i, v_i))}{\sqrt{\sum_{i=1}^{n} u_i^2} + \sqrt{\sum_{i=1}^{n} v_i^2}}$	
LinMod-6.2.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lagz_{t_x}(u_i, v_i)}{\sqrt{\sum_{i=1}^{n} u_i^2} + \sqrt{\sum_{i=1}^{n} v_i^2}}$	
LinMod-6.2.2.x	$s(u,v) = \frac{t_x^{-1}(\sum_{i=1}^{n} lagz_{t_x}(u_i, v_i))}{\sqrt{\sum_{i=1}^{n} u_i^2} + \sqrt{\sum_{i=1}^{n} v_i^2}}$	
LinMod-7.1.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lanz_{t_x}(u_i, v_i)}{\sqrt{\sum_{i=1}^{n} u_i^2} * \sqrt{\sum_{i=1}^{n} v_i^2}}$	
LinMod-7.1.2.x	$s(u,v) = \frac{t_x^{-1}(\sum_{i=1}^{n} lanz_{t_x}(u_i, v_i))}{\sqrt{\sum_{i=1}^{n} u_i^2} * \sqrt{\sum_{i=1}^{n} v_i^2}}$	
LinMod-7.2.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lagz_{t_x}(u_i, v_i)}{\sqrt{\sum_{i=1}^{n} u_i^2} * \sqrt{\sum_{i=1}^{n} v_i^2}}$	
LinMod-7.2.2.x	$s(u,v) = \frac{t_x^{-1}(\sum_{i=1}^{n} lagz_{t_x}(u_i, v_i))}{\sqrt{\sum_{i=1}^{n} u_i^2} * \sqrt{\sum_{i=1}^{n} v_i^2}}$	
LinMod-8.1.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lanz_{t_x}(u_i, v_i)}{\sum_{i=1}^{n} t_x(u_i) + \sum_{i=1}^{n} t_x(v_i)}$	
LinMod-8.1.2.x	$s(u,v) = \frac{t_x^{-1}(\sum_{i=1}^{n} lanz_{t_x}(u_i, v_i))}{t_x^{-1}(\sum_{i=1}^{n} t_x(u_i)) + t_x^{-1}(\sum_{i=1}^{n} t_x(v_i))}$	
LinMod-8.2.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lagz_{t_x}(u_i, v_i)}{\sum_{i=1}^{n} t_x(u_i) + \sum_{i=1}^{n} t_x(v_i)}$	
LinMod-8.2.2.x	$s(u,v) = \frac{t_x^{-1}(\sum_{i=1}^{n} lagz_{t_x}(u_i, v_i))}{t_x^{-1}(\sum_{i=1}^{n} t_x(u_i)) + t_x^{-1}(\sum_{i=1}^{n} t_x(v_i))}$	
LinMod-9.1.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lanz_{t_x}(u_i, v_i)}{\sum_{i=1}^{n} t_x(u_i) * \sum_{i=1}^{n} t_x(v_i)}$	
LinMod-9.1.2.x	$s(u,v) = \frac{t_x^{-1}(\sum_{i=1}^{n} lanz_{t_x}(u_i, v_i))}{t_x^{-1}(\sum_{i=1}^{n} t_x(u_i)) * t_x^{-1}(\sum_{i=1}^{n} t_x(v_i))}$	
LinMod-9.2.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lagz_{t_x}(u_i, v_i)}{\sum_{i=1}^{n} t_x(u_i) * \sum_{i=1}^{n} t_x(v_i)}$	
LinMod-9.2.2.x	$s(u,v) = \frac{t_x^{-1}(\sum_{i=1}^{n} lagz_{t_x}(u_i, v_i))}{t_x^{-1}(\sum_{i=1}^{n} t_x(u_i)) * t_x^{-1}(\sum_{i=1}^{n} t_x(v_i))}$	

	Definition	Reference
LinHRMod-1.1.1.x	$s(u,v) = \sum_{i=1}^{n} lmnz_{t_x}(u_i, v_i)$	
LinHRMod-1.1.2.x	$s(u,v) = t_x^{-1}\left(\sum_{i=1}^{n} lmnz_{t_x}(u_i, v_i)\right)$	
LinHRMod-1.2.1.x	$s(u,v) = \sum_{i=1}^{n} lmgz_{t_x}(u_i, v_i)$	
LinHRMod-1.2.2.x	$s(u,v) = t_x^{-1}\left(\sum_{i=1}^{n} lmgz_{t_x}(u_i, v_i)\right)$	
LinHRMod-2.1.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lmnz_{t_x}(u_i, v_i)}{\sum_{i=1}^{n} \lvert t_x(u_i)\rvert + \sum_{i=1}^{n} \lvert t_x(v_i)\rvert}$	
LinHRMod-2.1.2.x	$s(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} lmnz_{t_x}(u_i, v_i)\right)}{t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(u_i)\rvert\right) + t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(v_i)\rvert\right)}$	
LinHRMod-2.2.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lmgz_{t_x}(u_i, v_i)}{\sum_{i=1}^{n} \lvert t_x(u_i)\rvert + \sum_{i=1}^{n} \lvert t_x(v_i)\rvert}$	
LinHRMod-2.2.2.x	$s(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} lmgz_{t_x}(u_i, v_i)\right)}{t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(u_i)\rvert\right) + t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(v_i)\rvert\right)}$	
LinHRMod-3.1.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lmnz_{t_x}(u_i, v_i)}{\sum_{i=1}^{n} \lvert t_x(u_i)\rvert * \sum_{i=1}^{n} \lvert t_x(v_i)\rvert}$	
LinHRMod-3.1.2.x	$s(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} lmnz_{t_x}(u_i, v_i)\right)}{t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(u_i)\rvert\right) * t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(v_i)\rvert\right)}$	
LinHRMod-3.2.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lmgz_{t_x}(u_i, v_i)}{\sum_{i=1}^{n} \lvert t_x(u_i)\rvert * \sum_{i=1}^{n} \lvert t_x(v_i)\rvert}$	
LinHRMod-3.2.2.x	$s(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} lmgz_{t_x}(u_i, v_i)\right)}{t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(u_i)\rvert\right) * t_x^{-1}\left(\sum_{i=1}^{n} \lvert t_x(v_i)\rvert\right)}$	
LinHRMod-4.1.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lmnz_{t_x}(u_i, v_i)}{\sqrt{\sum_{i=1}^{n} t_x(u_i)^2} + \sqrt{\sum_{i=1}^{n} t_x(v_i)^2}}$	
LinHRMod-4.1.2.x	$s(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} lmnz_{t_x}(u_i, v_i)\right)}{\sqrt{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i)^2\right)} + \sqrt{t_x^{-1}\left(\sum_{i=1}^{n} t_x(v_i)^2\right)}}$	
LinHRMod-4.2.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lmgz_{t_x}(u_i, v_i)}{\sqrt{\sum_{i=1}^{n} t_x(u_i)^2} + \sqrt{\sum_{i=1}^{n} t_x(v_i)^2}}$	
LinHRMod-4.2.2.x	$s(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} lmgz_{t_x}(u_i, v_i)\right)}{\sqrt{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i)^2\right)} + \sqrt{t_x^{-1}\left(\sum_{i=1}^{n} t_x(v_i)^2\right)}}$	
LinHRMod-5.1.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lmnz_{t_x}(u_i, v_i)}{\sqrt{\sum_{i=1}^{n} t_x(u_i)^2} * \sqrt{\sum_{i=1}^{n} t_x(v_i)^2}}$	
LinHRMod-5.1.2.x	$s(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} lmnz_{t_x}(u_i, v_i)\right)}{\sqrt{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i)^2\right)} * \sqrt{t_x^{-1}\left(\sum_{i=1}^{n} t_x(v_i)^2\right)}}$	
LinHRMod-5.2.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lmgz_{t_x}(u_i, v_i)}{\sqrt{\sum_{i=1}^{n} t_x(u_i)^2} * \sqrt{\sum_{i=1}^{n} t_x(v_i)^2}}$	
LinHRMod-5.2.2.x	$s(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} lmgz_{t_x}(u_i, v_i)\right)}{\sqrt{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i)^2\right)} * \sqrt{t_x^{-1}\left(\sum_{i=1}^{n} t_x(v_i)^2\right)}}$	

	Definition	Reference
LinHRMod-6.1.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lmnz_{t_x}(u_i, v_i)}{\sqrt{\sum_{i=1}^{n} u_i^2} + \sqrt{\sum_{i=1}^{n} v_i^2}}$	
LinHRMod-6.1.2.x	$s(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} lmnz_{t_x}(u_i, v_i)\right)}{\sqrt{\sum_{i=1}^{n} u_i^2} + \sqrt{\sum_{i=1}^{n} v_i^2}}$	
LinHRMod-6.2.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lmgz_{t_x}(u_i, v_i)}{\sqrt{\sum_{i=1}^{n} u_i^2} + \sqrt{\sum_{i=1}^{n} v_i^2}}$	
LinHRMod-6.2.2.x	$s(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} lmgz_{t_x}(u_i, v_i)\right)}{\sqrt{\sum_{i=1}^{n} u_i^2} + \sqrt{\sum_{i=1}^{n} v_i^2}}$	
LinHRMod-7.1.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lmnz_{t_x}(u_i, v_i)}{\sqrt{\sum_{i=1}^{n} u_i^2} * \sqrt{\sum_{i=1}^{n} v_i^2}}$	
LinHRMod-7.1.2.x	$s(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} lmnz_{t_x}(u_i, v_i)\right)}{\sqrt{\sum_{i=1}^{n} u_i^2} * \sqrt{\sum_{i=1}^{n} v_i^2}}$	
LinHRMod-7.2.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lmgz_{t_x}(u_i, v_i)}{\sqrt{\sum_{i=1}^{n} u_i^2} * \sqrt{\sum_{i=1}^{n} v_i^2}}$	
LinHRMod-7.2.2.x	$s(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} lmgz_{t_x}(u_i, v_i)\right)}{\sqrt{\sum_{i=1}^{n} u_i^2} * \sqrt{\sum_{i=1}^{n} v_i^2}}$	
LinHRMod-8.1.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lmnz_{t_x}(u_i, v_i)}{\sum_{i=1}^{n} t_x(u_i) + \sum_{i=1}^{n} t_x(v_i)}$	
LinHRMod-8.1.2.x	$s(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} lmnz_{t_x}(u_i, v_i)\right)}{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i)\right) + t_x^{-1}\left(\sum_{i=1}^{n} t_x(v_i)\right)}$	
LinHRMod-8.2.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lmgz_{t_x}(u_i, v_i)}{\sum_{i=1}^{n} t_x(u_i) + \sum_{i=1}^{n} t_x(v_i)}$	
LinHRMod-8.2.2.x	$s(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} lmgz_{t_x}(u_i, v_i)\right)}{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i)\right) + t_x^{-1}\left(\sum_{i=1}^{n} t_x(v_i)\right)}$	
LinHRMod-9.1.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lmnz_{t_x}(u_i, v_i)}{\sum_{i=1}^{n} t_x(u_i) * \sum_{i=1}^{n} t_x(v_i)}$	
LinHRMod-9.1.2.x	$s(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} lmnz_{t_x}(u_i, v_i)\right)}{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i)\right) * t_x^{-1}\left(\sum_{i=1}^{n} t_x(v_i)\right)}$	
LinHRMod-9.2.1.x	$s(u,v) = \frac{\sum_{i=1}^{n} lmgz_{t_x}(u_i, v_i)}{\sum_{i=1}^{n} t_x(u_i) * \sum_{i=1}^{n} t_x(v_i)}$	
LinHRMod-9.2.2.x	$s(u,v) = \frac{t_x^{-1}\left(\sum_{i=1}^{n} lmgz_{t_x}(u_i, v_i)\right)}{t_x^{-1}\left(\sum_{i=1}^{n} t_x(u_i)\right) * t_x^{-1}\left(\sum_{i=1}^{n} t_x(v_i)\right)}$	

APPENDIX B

A list of the most important weighting schemes tested

To follow the notations of Curran (2004), we denote word-feature pairs as (w,r,w') triplets. However, please note that dispite the style of our notations, our calculations and formulas rather follow Lin (1998a), as we found them more intuitive. Substituting any of the components with an * results in a set of triplets, where the * takes the value of all the possible elements of the given type. Frequency counts are denoted as f(w,r,w'), from which probabilities (p) are calculated as dividing by f(*,r,*). Further, type frequencies (n) are calculated by counting the number of elements in the set defined by the triplet, and the number of words and features having positive type frequencies are denoted as $N_{(w,r)}$ and $N_{(r,w')}$, respectively:

$$\begin{aligned} f(w,r,*) &= \sum_{w'} f(w,r,w') \\ f(*,r,w') &= \sum_{w} f(w,r,w') \\ f(*,r,*) &= \sum_{w,w'} f(w,r,w') \end{aligned} \tag{B.1}$$

$$\begin{aligned} p(w,r,w') &= \frac{f(w,r,w')}{f(*,r,*)} \\ p(w,r,*) &= \frac{f(w,r,*)}{f(*,r,*)} \\ p(*,r,w') &= \frac{f(*,r,w')}{f(*,r,*)} \end{aligned} \tag{B.2}$$

$$\begin{aligned} n(w,r,*) &= |f(w,r,*)| \\ n(*,r,w') &= |f(*,r,w')| \end{aligned} \tag{B.3}$$

$$\begin{aligned} N_{(w,r)} &= |(w,r)|n(w,r,*) > 0| \\ N_{(r,w')} &= |(r,w')|n(*,r,w') > 0| \end{aligned} \tag{B.4}$$

To avoid unnecessary special cases in the formulas, the following two simplifications have been used in all measures: $\frac{0}{0} = 0$ and $0 * log(0) = 0$. Further, to make the formulas simpler, in case of some information theoretic measures the standard contingency table notations are used Evert (2005):

$$
\begin{aligned}
N &= f(*, r, *) \\
O_{11} &= f(w, r, w') \\
O_{12} &= f(w, r, *) - f(w, r, w') \\
O_{21} &= f(*, r, w') - f(w, r, w') \\
O_{22} &= N - f(w, r, *) - f(*, r, w') \\
&\quad + f(w, r, w')
\end{aligned}
\tag{B.5}
$$

There are such weighting schemes that could return a non-zero weight even if $f(w, r, w') = 0$ (for example due to the use of smoothing or the calculation of the logarithm of this value). However, in most cases it would not be beneficial to calculate and use these values (for example $log(0) = -\infty$), so in case of $f(w, r, w') = 0$ we have always taken the corresponding weight to be 0 too.

There are possible suffixes to the weighting schemes that work the following way:

- Tc0: No transformation on the counts
- Tc1: the same transformation on the counts as in PmiAl
- Tc2: the same transformation on the counts as in PmiDisc
- Tc3: the same transformation on the counts as in PmiWls
- Tc4: the same transformation on the counts as in Unis
- Tw0: no transformation on the weights
- Tw1: the same transformation on the weights as in WPmi9

- Tw2: the same transformation on the weights as in WPmi10
- Tw3: the same transformation on the weights as in WPmi7
- Tw4: multiplication of the weights with the multiplier in WPmi7
- S0: no subtraction from the weights
- S1: the same subtraction from the weights as in SPmi
- S2: the same subtraction from the weights as in Unis
- P0: no multiplication of the weight
- P1: the same multiplication of the weight as in TfIdf2
- P2: the same multiplication of the weight as in TfIdf7
- P3: the same multiplication of the weight as in WPmi7
- P4: the same multiplication of the weight as in PmiWdf
- P5: the same multiplication of the weight as in NPmi

	Definition	Reference
ATC	$\frac{0.5 + 0.5 * \frac{f(w,r,w')}{max(f(*,r,*))} * lb\frac{f(*,r,*)}{n(*,r,w')}}{\sqrt{\sum_w (a_1 * lb\,(a_2))^2}}$ $a_1 = 0.5 + 0.5 * \frac{f(w,r,w')}{max\,(f(*,r,*))}$ $a_2 = \frac{f(*,r,*)}{n(*,r,w')}$	(Kiela and Clark, 2014)
Chi^2	$\frac{N(O_{11} * O_{22} - O_{12} * O_{21})^2}{(O_{11}+O_{12})(O_{11}+O_{21})(O_{12}+O_{22})(O_{21}+O_{22})}$	(Evert, 2008)
Chi^{2}WYCC		(Evert, 2008)
Dice	$\frac{2 * f(w,r,w')}{f(w,r,*) + f(*,r,w')}$	(Curran, 2004)
Freq	$f(w,r,w')$	(Jurafsky and Martin, 2009)
GMean	$\frac{O_{11}}{\sqrt{O_{11} + O_{12} * (O_{12}+O_{22})}}$	(Evert, 2005)
Gref1	$\frac{lb\,(1 + f(w,r,w'))}{lb\,(1 + n(*,r,w'))}$	(Curran, 2004)
Gref2	$\frac{lb\,(1 + f(w,r,w'))}{1 + \sum_w \frac{f(w,r,w')}{f(*,r,w')} * lb\frac{f(w,r,w')}{f(*,r,w')}}$	(Curran, 2004)
Identity	$sgn\,(f(w,r,w'))$	(Curran, 2004)
Jaccard	$\frac{O_{11}}{O_{11}+O_{12}+O_{21}}$	(Evert, 2005)
JointProb	$\frac{f(w,r,w')}{f(*,r,*)}$	(Jurafsky and Martin, 2009)
Liddell	$\frac{O_{11} * O_{22} - O_{12} * O_{21}}{(O_{11}+O_{21}) * (O_{12}+O_{22})}$	(Evert, 2005)

	Definition	**Reference**
Lin1	$-log\frac{n(*,r,w')}{N_{(w,r)}}$	(Lin, 1998a)
Lin2	$-log\frac{n(*,r,w')}{f(*,r,*)}$	(Kiela and Clark, 2014)
Lin3	$-log\left(f(w,r,w')\right) * log\frac{n(*,r,w')}{N_{(w,r)}}$	(Kiela and Clark, 2014)
LogFreq	$lb\left(1+f(w,r,w')\right)$	
LogLHR	$-2log\left(\frac{L(c_{12},c_1,p)*L(c_2-c_{12},N-c_1,p)}{L(c_{12},c_1,p_1)*L(c_2-c_{12},N-c_1,p_2)}\right)$ $c_1=(w,r,*);\ c_2=f(*,r,w')$ $c_{12}=f(w,r,w');\ N=f(*,r,*)$ $p=\frac{c_2}{N};\ p_1=\frac{c_{12}}{c_1};\ p_2=\frac{c_2-c_{12}}{N-c_1}$ $L(k,n,x)=x^k*(1-x)^{n-k}$	(Evert, 2005)
LPMI1	$f(w,r,w')*lb\left(\frac{f(*,r,*)*f(w,r,w')}{f(w,r,*)*f(*,r,w')}\right)$	(Evert, 2005)
LPMI2	$\frac{f(w,r,w')}{f(*,r,*)}*lb\left(\frac{f(*,r,*)*f(w,r,w')}{f(w,r,*)*f(*,r,w')}\right)$	(Herger, 2014)
LTU	$\frac{(lb\left(f(w,r,w')\right)+1)*lb\left(\frac{f(*,r,*)}{n(*,r,w')}\right)}{0.8+0.2*\frac{f(*,r,w')}{avg(f(*,r,w'))}}$	(Reed et al., 2006)
MinPMITTest1	$min(PMI,TTest1)$	(Evert, 2005)
MinPMITTest2	$min(PMI,TTest2)$	(Evert, 2005)
MinSens	$min\left(\frac{O_{11}}{O_{11}+O_{12}},\frac{O_{11}}{O_{11}+O_{21}}\right)$	(Evert, 2005)
NPMI	$\frac{lb\left(\frac{f(*,r,*)*f(w,r,w')}{f(w,r,*)*f(*,r,w')}\right)}{NPMI_n}$ $NPMI_n=-lb\left(\frac{f(w,r,w')}{f(*,r,*)}\right)$	(Harispe et al., 2015)

	Definition	Reference
OddsRatio1	$log\frac{(O_{11}+0.5)*(O_{22}+0.5)}{(O_{12}+0.5)*(O_{21}+0.5)}$	(Lowe, 2001)
Okapi1	$p_1 * lb(p_2)$ $p_1 = \frac{f(w,r,w')}{0.5+0.5*\frac{f(*,r,w')}{avg(f(*,r,w'))}+f(w,r,w')}$ $p_2 = \frac{f(*,r,*)-n(*,r,w')+0.5}{f(w,r,w')+0.5}$	based on (Reed et al., 2006)
PLFFI	$lb\left(1+f(w,r,w')\right)*log\left(1-log\left(\frac{n(*,r,w')}{f(*,*,*)}\right)\right)$	basd on (Dobó and Csirik, 2013)
Rapp	$lb\left(1+\frac{f(w,r,w')}{f(*,r,w')}\right)*-\sum_w q*lb\left(q\right)$ $q = \frac{f(w,r,w')}{f(*,r,w')}$	(Rapp, 2003)
Rapp1	$lb\left(1+f(w,r,w')\right)*-\sum_w q*lb\left(q\right)$ $q = \frac{f(w,r,w')}{f(*,r,w')}$	based on Rapp
RelRisk1	$\frac{O_{11}*(O_{12}+O_{22})}{O_{12}*(O_{11}+O_{21})}$	(Sistrom and Garvan, 2004)
RelRisk2	$lb\frac{O_{11}*(O_{12}+O_{22})}{O_{12}*(O_{11}+O_{21})}$	(Evert, 2005)

	Definition	Reference
PMI	$lb\left(\frac{f(*,r,*)*f(w,r,w')}{f(w,r,*)*f(*,r,w')}\right)$	(Church and Hanks, 1990)
PMIAlpha	$lb\left(\frac{(\sum_{i=1}^{n} f(*,r,i)^{\alpha})*f(w,r,w')}{f(w,r,*)*f(*,r,w')^{\alpha}}\right)$ $\alpha = 0.75$	(Levy et al., 2015)
PMIAlphaWOLog	$\frac{(\sum_{i=1}^{n} f(*,r,i)^{\alpha})*f(w,r,w')}{f(w,r,*)*f(*,r,w')^{\alpha}}$ $\alpha = 0.75$	(Zhang et al., 2015)
PMICurran	$lb\left(\frac{f(*,*,*)*f(w,r,w')}{f(w,*,*)*f(*,r,w')}\right)$	(Curran, 2004)
PMIDisc	$lb\left(\frac{f(*,r,*)*(f(w,r,w')-disc)}{f(w,r,*)*f(*,r,w')}\right)$ $disc = 0.95$	(Lin, 1998b)
PMI^2	$lb\left(\frac{f(*,r,*)*f(w,r,w')^2}{f(w,r,*)*f(*,r,w')}\right)$	(Evert, 2005)
PMI^3	$lb\left(\frac{f(*,r,*)*f(w,r,w')^3}{f(w,r,*)*f(*,r,w')}\right)$	(Evert, 2005)
PMIWDF	$PMIWDF_{\delta} * lb\left(\frac{f(*,r,*)*f(w,r,w')}{f(w,r,*)*f(*,r,w')}\right)$ $PMIWDF_{\delta} = \frac{f(w,r,w')}{f(w,r,w')+1} * \frac{min\left(f(w,r,*), f(*,r,w')\right)}{min\left(f(w,r,*), f(*,r,w')\right)+1}$	(Pantel and Lin, 2002)
PMIWLS	$lb\left(\frac{(f(*,r,*)+1)*(f(w,r,w')+1)}{(f(w,r,*)+1)*(f(*,r,w')+1)}\right)$	(Turney and Pantel, 2010)
PMI*Chi^2		product of 2 weights
PMI*CondProb21		product of 2 weights
PMI*CondProb22		product of 2 weights
PMI*CondProb24		product of 2 weights
PMI*CondProb26		product of 2 weights

	Definition	Reference
PMI	$lb\left(\frac{f(*,r,*)*f(w,r,w')}{f(w,r,*)*f(*,r,w')}\right)$	(Church and Hanks, 1990)
PMI*Liddell		product of 2 weights
PMI*Lin11		product of 2 weights
PMI*Lin12		product of 2 weights
PMI*Lin13		product of 2 weights
PMI*Lin14		product of 2 weights
PMI*Lin15		product of 2 weights
PMI*Lin16		product of 2 weights
PMI*LTU		product of 2 weights
PMI*OddsRatio3		product of 2 weights
PMI*Okapi1		product of 2 weights
PMI*Rapp1		product of 2 weights
PMI*Rapp4		product of 2 weights
PMI*Rapp6		product of 2 weights
PMI*RelRisk1		product of 2 weights
PMI*RelRisk2		product of 2 weights
PMI*TFIDF1		product of 2 weights
PMI*Ttest1		product of 2 weights
PMI*Ttest2		product of 2 weights
SPMI	$lb\left(\frac{f(*,r,*)*f(w,r,w')}{f(w,r,*)*f(*,r,w')}\right) - SPMI_d$ $SPMI_d = lb(SPMI_k) \quad SPMI_k = 5$	(Weir et al., 2016)
SqLogLHR		based on (Pecina, 2010)
TCombCost		(Pecina, 2010)

	Definition	Reference
PMI	$lb\left(\frac{f(*,r,*)*f(w,r,w')}{f(w,r,*)*f(*,r,w')}\right)$	(Church and Hanks, 1990)
TFICF1	$lb\left(f(w,r,w')\right)*lb\left(\frac{f(*,r,*)}{f(*,r,w')}\right)$	(Kiela and Clark, 2014)
TFICF2	$lb\left(1+f(w,r,w')\right)*lb\left(1+\frac{f(*,r,*)}{f(*,r,w')}\right)$	based on (Kiela and Clark, 2014)
TFICF3	$lb\left(1+f(w,r,w')\right)*lb\left(\frac{1+f(*,r,*)}{1+f(*,r,w')}\right)$	(Reed et al., 2006)
TFIDF1	$\frac{f(w,r,w')}{n(*,r,w')}$	(Curran, 2004)
TFIDF2	$\frac{lb\left(1+f(w,r,w')\right)}{lb\left(1+\frac{N_{(r,w')}}{n(*,r,w')}\right)}$	(Curran, 2004)
TFIDF3	$f(w,r,w')*lb\left(1+\frac{f(*,r,*)}{f(*,r,w')}\right)$	(Jurafsky and Martin, 2009)
TFIDF4	$lb\left(f(w,r,w')\right)*lb\left(\frac{f(*,r,*)}{n(*,r,w')}\right)$	(Kiela and Clark, 2014)
TFIDF5	$lb\left(1+f(w,r,w')\right)*lb\left(1+\frac{f(*,r,*)}{n(*,r,w')}\right)$	(Kiela and Clark, 2014)
TTest1	$\frac{p(w,r,w')-p(w,r,*)*p(*,r,w')}{\sqrt{\frac{p(w,r,w')}{f(*,r,*)}}}$	(Weeds and Weir, 2005)
TTest2	$\frac{p(w,r,w')-p(w,r,*)*p(*,r,w')}{\sqrt{p(w,r,*)*p(*,r,w')}}$	(Jurafsky and Martin, 2009)
TTest3	$\frac{f(w,r,w')-\frac{f(w,r,*)*f(*,r,w')}{f(*,r,*)}}{\sqrt{f(w,r,w')*(1-\frac{f(w,r,w')}{f(*,r,*)})}}$	(Pecina, 2010)
UniSubtuples	$lb\left(\frac{O11*O22}{O12*O21}\right)-3.29*\sqrt{\frac{1}{O11}+\frac{1}{O12}+\frac{1}{O21}+\frac{1}{O22}}$	(Pecina, 2010)

	Definition	Reference
PMI	$lb\left(\frac{f(*,r,*)*f(w,r,w')}{f(w,r,*)*f(*,r,w')}\right)$	(Church and Hanks, 1990)
WPMI	$\frac{f(w,r,w')}{f(*,r,*)} * lb\left(\frac{f(*,r,*)*f(w,r,w')}{f(w,r,*)*f(*,r,w')}\right)$	(Fung and McKeown, 1997)
WPMI10	$lb\left(1+\frac{f(*,r,*)*f(w,r,w')}{f(w,r,*)*f(*,r,w')}\right)^2$	based on WPMI
WPMI4	$lb\left(1+f(w,r,w')\right)*PMI$	
ZTest1	$\frac{p(w,r,w')-p(w,r,*)*p(*,r,w')}{\sqrt{\frac{p(w,r,*)*p(*,r,w')}{f(*,r,*)}}}$	(Weeds and Weir, 2005)
ZTest2	$\frac{f(w,r,w')-\frac{f(w,r,*)*f(*,r,w')}{f(*,r,*)}}{\sqrt{\frac{f(w,r,*)*f(*,r,w')}{f(*,r,*)}}}$	(Evert, 2005)
ZTest3	$\frac{f(w,r,w')-\frac{f(w,r,*)*f(*,r,w')}{f(*,r,*)}}{\sqrt{\frac{f(w,r,*)*f(*,r,w')}{f(*,r,*)}*(1-\frac{f(w,r,*)*f(*,r,w')}{f(*,r,*)^2})}}$	(Pecina, 2010)
PMIAlpha	$lb\left(\frac{(\sum_{i=1}^{n} f(*,r,i)^{\alpha})*f(w,r,w')}{f(w,r,*)*f(*,r,w')^{\alpha}}\right)$ $\alpha = 0.75$	(Levy et al., 2015)
PMIWDF	$PMIWDF_{\delta} * lb\left(\frac{f(*,r,*)*f(w,r,w')}{f(w,r,*)*f(*,r,w')}\right)$ $PMIWDF_{\delta} = \frac{f(w,r,w')}{f(w,r,w')+1} * \frac{min\left(f(w,r,*),f(*,r,w')\right)}{min\left(f(w,r,*),f(*,r,w')\right)+1}$	(Pantel and Lin, 2002)
NPMI	$\frac{lb\left(\frac{f(*,r,*)*f(w,r,w')}{f(w,r,*)*f(*,r,w')}\right)}{NPMI_n}$ $NPMI_n = -lb\left(\frac{f(w,r,w')}{f(*,r,*)}\right)$	(Harispe et al., 2015)

	Definition	Reference
SPMI	$lb\left(\frac{f(*,r,*)*f(w,r,w')}{f(w,r,*)*f(*,r,w')}\right) - SPMI_d$ $SPMI_d = lb(SPMI_k) \quad SPMI_k = 5$	(Weir et al., 2016)
UniSubtuples	$lb\left(\frac{O11*O22}{O12*O21}\right) - 3.29*\sqrt{\frac{1}{O11}+\frac{1}{O12}+\frac{1}{O21}+\frac{1}{O22}}$	(Pecina, 2010)
NPMIALPHA	$\frac{PMIAlpha}{NPMI_n}$	
SPMIALPHA	$PMIAlpha - SPMI_d$	
PMIALPHAWDF	$PMIAlpha * PMIWDF_\delta$	
NSPMIALPHA	$\frac{PMIAlpha - SPMI_d}{NPMI_n}$	
NPMIALPHAWDF	$\frac{PMIAlpha * PMIWDF_\delta}{NPMI_n}$	
SPMIALPHAWDF	$(PMIAlpha - SPMI_d) * PMIWDF_\delta$	
NSPMIALPHAWDF	$\frac{(PMIAlpha - SPMI_d) * PMIWDF_\delta}{NPMI_n}$	
NSPMI	$\frac{PMI - SPMI_d}{NPMI_n}$	
NPMIWDF	$\frac{PMI * PMIWDF_\delta}{NPMI_n}$	
NSPMIWDF	$\frac{(PMI - SPMI_d) * PMIWDF_\delta}{NPMI_n}$	
SPMIWDF	$(PMI - SPMI_d) * PMIWDF_\delta$	
PMIALPHAUNISUBT	$lb\left(\frac{O11*O22_\alpha}{O12*O21_\alpha}\right) - 3.29*\sqrt{\frac{1}{O11}+\frac{1}{O12}+\frac{1}{O21_\alpha}+\frac{1}{O22_\alpha}}$ $O22_\alpha = \left(\sum_{i=1}^{n} f(*,r,i)^\alpha\right) - f(w,r,*) - f(*,r,i)^\alpha + f(w,r,w')$ $O21_\alpha = f(*,r,i)^\alpha - f(w,r,w') \quad \alpha = 0.75$	
NPMIUNISUBT	$\frac{UniSubtuples}{NPMI_n}$	
SPMIUNISUBT	$UniSubtuples - SPMI_d$	

	Definition	Reference
PMIWDFUNISUBT	$UniSubtuples * PMIWDF_{\delta}$	
NPMIALPHAUNISUBT	$\frac{PMIALPHAUNISUBT}{NPMI_n}$	
SPMIALPHAUNISUBT	$PMIALPHAUNISUBT - SPMI_d$	
PMIALPHAWDFUNISUBT	$PMIALPHAUNISUBT * PMIWDF_{\delta}$	
NSPMIALPHAUNISUBT	$\frac{PMIALPHAUNISUBT - SPMI_d}{NPMI_n}$	
NPMIALPHAWDFUNISUBT	$\frac{PMIALPHAUNISUBT * PMIWDF_{\delta}}{NPMI_n}$	
SPMIALPHAWDFUNISUBT	$(PMIALPHAUNISUBT - SPMI_d) * PMIWDF_{\delta}$	
NSPMIALPHAWDFUNISUBT	$\frac{(PMIALPHAUNISUBT - SPMI_d) * PMIWDF_{\delta}}{NPMI_n}$	
NSPMIUNISUBT	$\frac{UNISUBT - SPMI_d}{NPMI_n}$	
NPMIWDFUNISUBT	$\frac{UNISUBT * PMIWDF_{\delta}}{NPMI_n}$	
NSPMIWDFUNISUBT	$\frac{(UNISUBT - SPMI_d) * PMIWDF_{\delta}}{NPMI_n}$	
SPMIWDFUNISUBT	$(UNISUBT - SPMI_d) * PMIWDF_{\delta}$	
PMIALPHAUNISUBTAM	$\frac{PMIALPHA + UNISUBT}{2}$	
PMIALPHAUNISUBTGM	$\sqrt{PMIALPHA * UNISUBT}$	
PMIALPHAUNISUBTHM	$\frac{2 * PMIALPHA * UNISUBT}{PMIALPHA + UNISUBT}$	
PMIALPHAUNISUBTPROD	$PMIALPHA * UNISUBT$	
PMIALPHAUNISUBTLOGPROD	$t_{lb}(PMIALPHA) * t_{lb}(UNISUBT)$	
NPMIALPHAAM	$\frac{NPMI + PMIALPHA}{2}$	
NPMIALPHAGM	$\sqrt{NPMI * PMIALPHA}$	
NPMIALPHAHM	$\frac{2 * NPMI * PMIALPHA}{NPMI + PMIALPHA}$	
NPMIALPHAPROD	$NPMI * PMIALPHA$	
NPMIALPHALOGPROD	$t_{lb}(NPMI) * t_{lb}(PMIALPHA)$	

APPENDIX C

The used Hungarian datasets

C.1 Hungarian TOEFL dataset part 1

érkezés | eljövetel | letartóztatás | finanszírozás | stabilitás

évkönyv | krónikák | otthonok | nyomok | dalok

vezetés | hatalom | megfigyelés | szerelem | tudatosság

vita | veszekedés | háború | választás | verseny

eltérések | különbségek | súlyok | betétek | hullámhosszak

mód | módszer | fejadag | öl | őrület

színárnyalat | szín | ragyogás | kontraszt | illat

témák | tárgyak | képzés | fizetések | előnyök

százalék | arány | térfogat | minta | profit

elrendezett | megtervezett | megmagyarázott | tanult | eldobott

épített | konstruált | előterjesztett | finanszírozott | szervezett

kiagyalt | kieszelt | kitisztított | kért | felügyelt

elfogyasztott | megevett | nevelt | elfogott | ellátott

szertefoszlik | eloszlik | elszigetel | álcáz | lefényképez

forgalmaz | terjeszt | elüzletiesít | kutat | elismer

kiszámol | megold | felsorol | feloszt | kifejez

ad | szállít | lenyűgöz | megvéd | tanácsol

vigyorog | mosolyog | edz | pihen | viccel

üdvözölt | köszöntött | ítélt | emlékezett | címzett

nagyobb | tekintélyesebb | állandóbb | közelebbi | jobb

fesztelen | közvetlen | megörökített | félreértett | helytelen

költséges | drága | szép | népszerű | bonyolult

lezser | nyugodt | határmenti | unalmas | gazdasági

képzeletbeli | fantáziadús | megszokott | nyilvánvaló | logikus

megvalósítható | lehetséges | megengedett | igazságos | nyilvánvaló

hibás | tökéletlen | apró | fénylő | durva

hátulsó | hátsó | görbe | izmos | szőrös

végtelen | korlátlan | viszonylagos | szokatlan | szerkezeti

hegyes | éles | hasznos | egyszerű | híres

keskeny | vékony | tiszta | fagyos | mérgező

veszedelmes | veszélyes | kötelező | izgalmas | sértő

tömören | röviden | erőteljesen | pozitívan | szabadon

állandóan | folyamatosan | azonnal | gyorsan | véletlenül

ügyesen | szakképzetten | megfontoltan | alkalmanként | humorosan

roppantul | borzasztóan | helyénvalóan | egyedülállóan | kétségkívül

lényegében | alapvetően | talán | mohón | átlagosan

gyorsan | sebesen | gyakran | valójában | ismételten

általánosan | nagyjából | leíróan | vitatottan | pontosan

kedvetlenül | közönyösen | szokásszerűen | kétpártilag | rendhagyóan

összevissza | véletlenszerűen | veszélyesen | sűrűn | lineárisan

C.2 Hungarian TOEFL dataset part 2

kitartás | tartósság | képesség | nagylelkűség | zavar

orvos | doktor | vegyész | gyógyszerész | ápolónő

feltételek | kikötések | kapcsolatok | hatáskörök | értelmezések

gyökerek | eredetek | szertartások | gyógymód | funkció

üdvözlések | köszöntések | információ | ceremóniák | kiváltságok

hely | helyszín | éghajlat | szélesség | tenger

feladatok | tennivalók | vásárlók | anyagok | boltok

nyugodtság | békesség | kíméletlenség | kimerültség | boldogság

csúcspont | tetőpont | befejezés | kezdet | hanyatlás

siettet | meggyorsít | megenged | meghatároz | elkísér

kiemel | kihangsúlyoz | módosít | utánoz | visszaállít

kiszabott | kirótt | hitt | kért | korrelált

szerez | kap | nyomtat | kereskedik | kölcsönvesz

értékesített | eladott | fagyasztott | édesített | hígított

megoldott | elintézett | közzétett | elfelejtett | megvizsgált

bemutatott | demonstrált | publikált | megismételt | elhalasztott

elhelyezett | pozícionált | forog | elszigetelt | kiürít

fenntartott | meghosszabbított | finomított | csökkentett | elemzett

befejeződött | végződött | beállított | elhalasztott | kiértékelt

elsődleges | fő | legtöbb | számos | kivételes

termékeny | eredményes | komoly | hozzáértő | ígéretes

kiemelkedő | kitűnő | kopott | antik | rejtélyes

leendő | potenciális | bizonyos | megfontolt | kiemelkedő

elismert | elfogadott | sikeres | ábrázolt | üdvözölt

kirívó | feltűnő | tüskés | szórakoztató | véletlen

magányos | egyedüli | éber | nyugtalan | rettenthetetlen

elegendő | elég | minapi | élettani | értékes

mérsékelt | enyhe | hideg | rövid | szeles

egyforma | hasonló | kemény | összetett | éles

valószínűtlen | esélytelen | barátságtalan | különböző | népszerűtlen

páratlan | egyedülálló | ismeretlen | elidegenített | felülmúlt

sietősen | sietve | agyafúrtan | szokásszerűen | időrendben

normálisan | általában | nehézkesen | maradandóan | időszakosan

gyakran | sűrűn | feltétlenül | vegyileg | alig

különösen | páratlanul | részben | hazafiasan | gyanakodva

elsősorban | főként | alkalmanként | óvatosan | következetesen

lassan | fokozatosan | ritkán | hatékonyan | folyamatosan

sürgősen | kétségbeesetten | tipikusan | elképzelhetően | próbaképpen

verbálisan | szóban | nyíltan | megfelelően | hosszadalmasan

vadul | dühösen | jellegzetesen | rejtélyesen | hirtelen

C.3 Hungarian Rubenstein-Goodenough dataset

elmegyógyintézet	sírkert	0.79
elmegyógyintézet	gyümölcs	0.19
elmegyógyintézet	bolondokháza	3.04
elmegyógyintézet	pap	0.39
aláírás	vízpart	0.06
aláírás	kézjegy	3.59
személygépkocsi	autó	3.92
személygépkocsi	párna	0.97
személygépkocsi	varázsló	0.11
madár	kakas	2.63
madár	daru	2.63
madár	erdőség	1.24
fiú	srác	3.82
fiú	baromfi	0.44
fiú	bölcs	0.96
báty	srác	2.41
báty	pap	2.74
autó	utazás	1.55
sírkert	temető	3.88
sírkert	földhalom	1.69
sírkert	erdőség	1.18
part	erdő	0.85
part	hegy	1.26
part	vízpart	3.60
kakas	baromfi	3.68
fonal	mosoly	0.02
fonal	kötél	3.41
daru	szerszám	2.37
daru	baromfi	1.41
párna	ékkő	0.45
párna	kispárna	3.84
étel	gyümölcs	2.69
étel	baromfi	1.09

erdő	temető	1.00
erdő	erdőség	3.65
gyümölcs	kemence	0.05
kemence	szerszám	1.37
kemence	tűzhely	3.11
drágakő	ékkő	3.94
üveg	ékkő	1.78
üveg	bűvész	0.44
üveg	ivópohár	3.45
temető	bolondokháza	0.42
vigyor	szerszám	0.18
vigyor	srác	0.88
vigyor	mosoly	3.46
hegy	földhalom	3.29
hegy	erdőség	1.48
szerszám	eszköz	3.66
utazás	út	3.58
srác	varázsló	0.99
bűvész	jós	1.82
bűvész	varázsló	3.21
délidő	dél	3.94
pap	jós	0.91
pap	rabszolga	0.57
földhalom	vízpart	0.97
földhalom	tűzhely	0.14
dél	kötél	0.04
jós	bölcs	2.61
baromfi	út	0.04
bölcs	varázsló	2.46
jobbágy	rabszolga	3.46
vízpart	út	1.22
vízpart	erdőség	0.90