# A study of model parameters for scaling up word to sentence similarity tasks in distributional semantics

Dmitrijs Milajevs

Submitted in partial fulfillment of the requirements of the Degree of Doctor of Philosophy

Queen Mary
University of London

March 9, 2018

# Statement of originality

I, Dmitrijs Milajevs, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of this thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

March 9, 2018

# Details of collaboration and publications

Content from the following publications appears in this thesis, which has been written with the guidance of my supervisors Mehrnoosh Sadrzadeh and Matthew Purver. Some of the publications were written in collaboration with Dimitri Kartsaklis, Thomas Roelleke and Sascha Griffiths. This thesis was proof read for the purposes of spelling and grammar by Sara Dyck.

1. Milajevs and Purver (2014). Main author. Investigating the Contribution of Distributional Semantic Information for Dialogue Act Classification. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC).*

2. Milajevs, Kartsaklis, Sadrzadeh, and Purver (2014). Main author. Evaluating Neural Word Representations in Tensor-Based Compositional Settings. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).*

3. Milajevs, Sadrzadeh, and Roelleke (2015). Main author. IR Meets NLP: On the Semantic Similarity Between Subject-Verb-Object Phrases. *In Proceedings of the 2015 International Conference on Theory of Information Retrieval.*

4. Milajevs, Sadrzadeh, and Purver (2016). Main author. Robust Co-occurrence Quantification for Lexical Distributional Semantics. *In Proceedings of the ACL 2016 Student Research Workshop.*

5. Milajevs and Griffiths (2016). Main author. A Proposal for Linguistic Similarity Datasets Based on Commonality Lists. *In Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP.*

# Software

The following software was developed as a part of this thesis.

- Fowler.corpora is an implementation of distributional lexical and sentential similarity models.

- Google-ngram-downloader is an on-the-fly reader of the Google Books ngram dataset.

- NLTK was extended in several ways. The BNC reader was extended to support the full BNC edition. Lesk's word sense disambiguation algorithm was extended to support part of speech tags. The Dependency Graph datastructure was refactored to support custom dependency graph construction. An interface to the Stanford CoreNLP web API was added to the NLTK.

This work was supported by the EPSRC grant EP/J002607/1.

# A study of model parameters for scaling up word to sentence similarity tasks in distributional semantics

**Dmitrijs Milajevs**

## Abstract

Representation of sentences that captures semantics is an essential part of natural language processing systems, such as information retrieval or machine translation. The representation of a sentence is commonly built by combining the representations of the words that the sentence consists of. Similarity between words is widely used as a proxy to evaluate semantic representations. Word similarity models are well-studied and are shown to positively correlate with human similarity judgements.

Current evaluation of models of sentential similarity builds on the results obtained in lexical experiments. The main focus is how the lexical representations are used, rather than what they should be. It is often assumed that the optimal representations for word similarity are also optimal for sentence similarity. This work discards this assumption and systematically looks for lexical representations that are optimal for similarity measurement between sentences.

We find that the best representation for word similarity is not always the best for sentence similarity and vice versa. The best models in word similarity tasks perform best with additive composition. However, the best result on compositional tasks is achieved with Kronecker-based composition. There are representations that are equally good in both tasks when used with multiplicative composition.

The systematic study of the parameters of similarity models reveals that the more information lexical representations contain, the more attention should be paid to noise. In particular, the word vectors in models with the feature size at the magnitude of the vocabulary size should be sparse, but if a small number of context features is used then the vectors should be dense.

Given the right lexical representations, compositional operators achieve state-of-the-art performance, improving over models that use neural-word embeddings. To avoid overfitting, either several test datasets should be used or parameter selection should be based on parameters' average behaviours.

Submitted for the degree of Doctor of Philosophy

Queen Mary University of London

March 9, 2018

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Computers require specially designed programming languages to be controlled, despite the fact that computers play a crucial role in our lives. Ideally, the interaction with a computer should not be different from the interaction with a human. Computational linguistics is one of the fields that addresses this problem.

Computers need to understand human language in order to be controlled by people in a casual manner. However, different tasks require various levels of language understanding. For instance, even if one does not recognise or know the language of a piece of text in Figure 1.1a, one can tell that there are 39 words and that there is only one sentence. One can even argue that this is a piece of poetry and the first line is its title, basing the argument on the shape of the text.

The conclusions above require neither the complete understanding of the language nor the meaning of the text. The knowledge of the format that poems are written in and that texts—at least in some languages—consist of words separated by a space is enough. Moreover, knowing the letter distribution across all human languages, or having a list of words in them, one would conclude that the text is in Latvian. These conclusions can be gathered without knowing what the text is about and are currently successfully implemented by computers.

On the other hand, a task that asks for a list of associations—an essay or a painting inspired by a piece of text—demands a much better understanding of the text that requires a deeper knowledge of the language and greater familiarity with the culture. Luckily, nowadays these kinds of tasks are not expected to be completed by computers in day-to-day life because people generally enjoy doing these things themselves.

However, it is reasonable to ask a computer the following questions regarding the text: a) What is the text of Figure 1.1a about? b) What is the relationship between the texts in Figure 1.1a and Figure 1.1b? c) Is the content similar or identical? d) Where did the meeting take place? e) What poems are similar to this?

Jaunkundze ar sunīti

Un Vecrīgas šķērsielā, šaurā
kā vēstuļu kastītes sprauga,
kur troksnim un burzmai tik atbalss,
kur smaržo pēc darvas,
        dzelzs un pēc āboliem pagrabos sausos,
es satiku jaunkundzi –
glītu un veiklu kā mēle,
kā spēlējot vijoles lociņš.

(a)

Барышня с собачкой

В Старой Риге, на улице поперечной, узкой,
как щель в почтовый ящик,
в который проникают только отголоски шума, гама,
где запах дёгтя, ржавчины и яблок в сухих подвалах,
я встретил барышню –
красива и ловка - она - язык,
смычок, играющий на скрипке.

(b)

Young Woman with a Dog

On a narrow side-street in Riga's old quarter,
as though in a mailbox slot
where noise and hustle only echo,
and it smells of tar and steel
and apples kept in dry basements,

I met a young woman
attractive and active
as a tongue,
as a violin-bow playing.

(c)

**Figure 1.1:** Three pieces of written natural language. The text in Figure 1.1a is the beginning of the poem "Jaunkundze ar sunīti" by Aleksandrs Čaks (1996), Figure 1.1b is a Russian translation by Lora Trin (the text is available at `http://grafomanam.net/works/34812`), and Figure 1.1c is an English translation by Inara Cedrins (2013).

Text summarisation, machine translation, information extraction and information retrieval are branches of computational linguistics that provide methods for answering these questions. The questions above have a general property: all of them are about a certain aspect of the meaning of the text. Semantics is an area that studies meaning representation and thus, is necessary to solve these tasks.

While it is not completely known how meaning is represented in the human mind, it is argued that similarity between two events or objects is based on the way humans represent them (Hahn 2014). Similarity judgements are easy to collect. Many similarity datasets exist that serve as proxies for evaluation of computational models of meaning.

The distributional hypothesis of Harris (1954)—that semantically similar words tend to appear in similar contexts—stands behind distributional models of meaning. In Figure 1.1c, the side-street occurs with the words *slot*, *noise*, *hustle* and *smells*. Such a company of words starkly contrasts with the words used to describe the woman: she is *young*, *attractive* and *active*.

Moreover, the descriptive, neighbouring words of the side-street bring images of other things similar to it that are noisy and smell. At the same time, the descriptive terms of the woman fire in the mind attractive and active associations, making the difference between the side-street and the woman even stronger.

Distributional models of word meaning (also known as lexical models of meaning) are based on the co-occurrence statistics of words in a large collection of texts (Mikolov et al. 2013a;b;c, Turney and Pantel 2010). The main challenge is to use the co-occurrence statistics efficiently. Because, even though the word *and* appears in the neighbourhood of the word *side-street* in the poem, it is much less descriptive of the properties of the street than the word *slot*. Nowadays, lexical models are well-studied, and their estimates of the similarity between word pairs are very close to human judgements for the same task (Baroni et al. 2014b, Halawi et al. 2012, Levy et al. 2015).

The estimation of the similarity of multi-word expressions is currently an active research topic. In comparison to the lexical models, where data are plentiful, the main challenge is data sparsity. There are infinitely many multi-word expressions, and most of them appear only once in a corpus. Even if we take all the books on Earth and write down all the utterances that were said, most of the sentences encountered would appear only once.

The dominant solution to the data sparsity problem is to build a compositional representation of a multi-word expression; that is, the same way in which Lego pieces are assembled into vehicles, buildings and many other types of objects. One advantage of such an approach is that the methods for obtaining word representations can be reused. The bricks are there, the question is how to assemble them together.

The compositional models come in many flavours. Mitchell and Lapata (2010) propose a method that ignores the word order and any grammatical structure of an expression. Baroni et al. (2014a), Coecke et al. (2010) investigate how the grammatical structure can be taken into account. Several implementations of Coecke et al. (2010)'s theoretical proposal exist—see the work of Fried et al. (2015), Grefenstette and Sadrzadeh (2011a;b), Kartsaklis and Sadrzadeh (2014). Chapter 2 gives an overview of lexical representations, the methods of composition and evaluation.

Until now, the main focus of the evaluation of compositional similarity models was the compositional operators. The word representations are usually taken such that they are good in lexical tasks. The fact that there might be a dependency between the word representations and the compositional methods is mostly overlooked. It is assumed that the findings based on the lexical experiments also apply to the compositional models.

The goal of this thesis is to study the link between the lexical representations and the methods of composition for similarity estimation. Once the optimal lexical parameters are identified for all compositional operators, the operators can be compared in the most accurate way.

The goal is expressed in two research questions:

- What is the performance limit of distributional models of meaning?
- How do compositional operators and lexical representations affect one another?

To answer these questions, we perform a large-scale study of similarity models over several parameters. The parameters are split into three kinds: the similarity measure, the weighting scheme and the amount of information associated with every item.[1]

The similarity measure defines how similarity is computed given two representations. The weighting scheme serves two roles. First, it distinguishes informative co-occurrence information from uninformative. Second, the weighting scheme minimises the effect of noise in the co-occurrence data. The amount of information for distributional modes is how many distinct words are considered to be valid, neighbouring words. This usually varies from a few thousand

[1] Only a small class of distributional models is being studied, specifically count models (Baroni et al. 2014b) with no dimensionality reduction. The count models are shown to be related to more sophisticated methods such as word2vec (Levy et al. 2015, Mikolov et al. 2013a;b;c), making them more fruitful for initial research. Models based on dimensionality reduction and word2vec bring not only more parameters increasing the total space of parameter combinations to explore, but also require much more time and computational resources to be instantiated.

most frequent words to the whole vocabulary. The description of model parameters is given in Chapter 3.

Regarding the first research question, our systematic study of parameters reveals that the performances of count-based distributional models are competitive with the current state-of-the-art lexical similarity estimation methods and even outperform some of them in the compositional setting. Notably, we show an improvement over the predictive methods (Mikolov et al. 2013a;b;c).

To answer the second research question, we extensively test compositional models to identify the best lexical representations for composition (Chapters 6 and 7). We find that, indeed, there is a link between compositional operators and lexical representations.

By taking into account the dependency between compositional operators and lexical representations, we achieve state-of-the-art results with additive and multiplicative composition. By reusing the best lexical representations with categorical compositional operators (Coecke et al. 2010), we improve their performance. Moreover, we show that the optimal parameters to measure the similarity between words (Chapter 4) are different from the optimal parameters to measure similarity between phrases.

## 1.1 Structure of this thesis

**Chapter 2** A review of logical and distributional models of meaning, description of the current similarity datasets and an overview of the current state-of-the-art models.

**Chapter 3** The methodology for robust selection of similarity models, description of used model parameters and list of hypotheses.

**Chapter 4** Experiments on the lexical datasets: SimLex-999 and MEN.

**Chapter 5** Description of PhraseRel, a new phrase relevance dataset.

**Chapter 6** Experiments on three phrasal datasets: GS11, KS14 and PhraseRel.

**Chapter 7** Selection of the models based on all datasets and experiments with tensor-based compositional methods.

**Chapter 8** Conclusion of the thesis.

# Chapter 2

# Background: Similarity between words and phrases

## 2.1 The notion of similarity

Similarity is the degree of resemblance between two objects or events (Hahn 2014). It plays a crucial role in psychological theories of knowledge and behaviour, where similarity is used to explain such phenomena as classification and conceptualisation (Hahn and Chater 1997, Markman and Gentner 1996, Medin et al. 1993, Tversky 1977, Tversky and Hutchinson 1986).

*Fruit* is a *category* because it is a practical generalisation. Fruits are sweet and generally are desserts, so when one is presented with an unseen fruit, one can hypothesise that it is served toward the end of a dinner.

Generalisations are extremely powerful in describing a language, as well. The verb *runs* requires its subject to be singular. *Verb*, *subject* and *singular* are categories that are used to describe English grammar. When one encounters an unknown word and is told that it is a verb, one will immediately have an idea about how to use it, assuming that it is used similarly to other English verbs.

From a computational perspective, this motivates and guides the development of similarity components that are embedded into automatic systems that deal with natural language.

The information that the word *carpet* is similar in meaning to the word *mat* might be exploited by a language model in estimating the probability of the sentence *the cat sat on the carpet*, even if it did not occur in the corpus but *the cat sat on the mat* did (Bengio et al. 2006).

In Information Retrieval (IR), queries are expanded with related terms to increase the number of retrieved relevant documents. For example, if a user issues the query *lakes in Sweden*, the system might add related words to the query such as *lake*, *reservoir*, *river* or even *swim* so that documents that do not contain the word *lakes* are retrieved (Xu and Croft 1996).

A dependency parser might benefit from a generalisation about the part-of-speech tag of a word which did not occur in the training data, based on its occurrence pattern in a large corpus of documents from the web (Andreas and Klein 2014, Hermann and Blunsom 2013).

A dialogue act tagging system might require classification of an utterance based on its role in a dialogue, such as a question or an acknowledgement (Kalchbrenner and Blunsom 2013).

The examples show that similarity is a broad term that is task-dependent. An IR system needs to identify semantically similar (*lake*, *river*) and related (*lake*, *swim*) terms. A dependency parser benefits from the similarity of word usage. A language model exploits similarity in word meaning. A dialogue act tagging system relies on the similarity of the roles that the utterances play in discourse.

A computational model that estimates similarity need not only take into account the different flavours of the similarity relation, but also be able to measure similarity between pairs of words, between pairs of phrases or even between whole sentences, utterances or documents.

## 2.2 Representation for similarity measurement

According to Hahn (2014), “similarity is an essentially psychological notion, based on the way we represent objects, that is, the way they appear to us.” Since it is not yet known how objects are represented in the human mind, the computational way of object representation for similarity estimation has to be agreed upon. However, one needs to be extremely careful when the object representation is decided, as it is unavoidably connected to the *meaning of words in isolation*.

Frege discusses two conflicting principles of meaning (Janssen 2001). According to *the principle of compositionality*, isolated word meanings are the building blocks of sentence meanings:

> The meaning of a compound expression is a function of the meaning of its parts and the syntactic rule by which they are combined. (Janssen 2001)

However, according to *the principle of contextuality*, the word meaning in isolation is not defined:

> Never ask for the meaning of a word in isolation, but only in the context of a sentence. (Janssen 2001)

It is worth noting here that the measurement of similarity in isolation is also problematic because the number of features an entity has is infinite, and it is easy to show that two entities will always have an infinite number of common features, making the degree of resemblance undefined (Goodman 1972, Hahn and Chater 1997). For example, the tree next to my house is similar to my house because both of them are less than one kilometre in height; both of them are less than two kilometres in height; both of them are less than three kilometre in height; and so on.

To make similarity measurements possible, they have to be measured *under a given description* (Hahn 2014, Markman and Gentner 1996, Medin et al. 1993). Thus, similarity is always contextualised. In other words, similarity emerges only when the possible properties are weighted. In our example, the tree and the house are similar with respect to the colour: both of them are green. The height properties and other irrelevant properties are assigned zero weight, making the number of non-zero feature values finite.

Frege's principle of contextuality allows us to define the meaning of a word by identifying its contribution to the meaning of a sentence. Firth's (1957) famous quote that "you shall know a word by the company it keeps," suggests that the word meaning can be *modelled* as the combination of the meanings of its occurrences in sentences of a corpus. Note that this does not provide the absolute word meaning, but only its meaning relative to the corpus. This assumption is also supported by the distributional hypothesis of Harris (1954) that the differences of occurrences of two words quantify the difference in their relative meaning, but do not necessarily define the meaning.

Once the relative word meaning is accepted, compositionality can be used to obtain representations of phrases and sentences (Baroni et al. 2014a, Coecke et al. 2010, Dowty et al. 1980, Janssen 2016, Montague 1970).

We continue with an overview of the main ideas behind the word and phrase representations for similarity measurement. It is later followed by the discussion of the common empirical evaluation procedures and the strategies of obtaining reliable evaluation results.

## 2.3 Representation of words

In principle, we would like to capture the intuition that while *John* and *Mary* are distinct entities, they are rather similar to each other—because both of them are humans—and are dissimilar to *dog*, *pavement* or *idea*. However, we start with the logical word meaning representation that captures the fact that entities are distinct but does not provide the means to measure similarity.

### 2.3.1 Logical representations for inference

Formal semantics provides the means to infer[1] some piece of information from another. The main studied relation is the entailment of sentences, for example, *John swims in Åresjön* entails

*John swims in a Swedish lake*. To evaluate entailment, the sentences are converted to formulas. The words correspond to symbols in formal logic.

The individual word *Åresjön* corresponds to the symbol *Åresjön'*, which is mapped to the actual lake by the interpretation function $\mathcal{I}$.

One-place properties are seen as sets of individuals, so $\mathcal{I}(\textit{Swedish}')$ is a set that contains $\mathcal{I}(\textit{Åresjön}')$ and $\mathcal{I}(\textit{Väsman}')$, among many other Swedish entities.

*Swim'* is a two-place predicate that is represented as a set that contains the pairs between which the relation holds; so if John actually swims in Åresjön, then $\mathcal{I}(\textit{swim}')$ will contain the pair $(\mathcal{I}(\textit{John}'), \mathcal{I}(\textit{Åresjön}'))$.

While such formalism is very powerful for entailment detection between sentences, similarity measurement is problematic,[2] because there is no relation between atomic symbols: we only know that *Åresjön* and *Väsman* correspond to different entities in the universe, but know nothing about the resemblance between their properties.

### 2.3.2 Distributional representations for similarity

Distributional methods provide a way to measure the similarity between words. The representations are produced by exploiting Harris' (1954) intuition that similar words occur in similar contexts.

A common approach is to construct a vector space in which the dimensions correspond to contexts, which are usually other words (Turney and Pantel 2010). The components of the vector of a word can be calculated by taking the frequency with which the word co-occurred with the corresponding contexts within a predefined window in a corpus of interest. The similarity in meaning can be expressed via a suitable distance metric within the space.

| | philosophy | book | school |
|---|---|---|---|
| John | 4 | 60 | 59 |
| Mary | 0 | 10 | 22 |
| girl | 0 | 19 | 93 |
| boy | 0 | 12 | 146 |
| idea | 10 | 47 | 39 |

**Table 2.1:** Word co-occurrence frequencies extracted from the BNC

Table 2.1 shows five three-dimensional vectors for words *Mary*, *John*, *girl*, *boy* and *idea*. These are *target words*. The words *philosophy*, *book* and *school* label vector space dimensions and are referred to as *context words*. Table 2.1 represents the global co-occurrence statistics, giving the name to the representations.

[1] Work on natural logic demonstrates that it is not necessary to convert to a logical representations, see, for example, MacCartney and Manning (2007).

[2] Such property could be perfectly expressed in formal semantics, but it is not generally seen as part of the job of formal semantics, also acquiring such properties is hard in comparison to the distributional methods.

As the vector for *Mary* is closer to *girl* than it is to *boy* in the vector space, we can say that *Mary* shares more features with *girl* (and less with *boy*), therefore *Mary* is semantically more similar to *girl* than to *boy*.

Mathematically, the similarity can be expressed using, for instance, the cosine of the angle between two vectors:

$$\cos(\theta) = \frac{\overrightarrow{Mary} \cdot \overrightarrow{girl}}{||\overrightarrow{Mary}||||\overrightarrow{girl}||} = \frac{(0 \times 0) + (10 \times 19) + (22 \times 93)}{\sqrt{0^2 + 10^2 + 22^2}\sqrt{0^2 + 19^2 + 93^2}} \approx \frac{2236}{2294} \approx 0.975$$

$$\cos(\phi) = \frac{\overrightarrow{Mary} \cdot \overrightarrow{boy}}{||\overrightarrow{Mary}||||\overrightarrow{boy}||} = \frac{(0 \times 0) + (10 \times 12) + (22 \times 146)}{\sqrt{0^2 + 10^2 + 22^2}\sqrt{0^2 + 12^2 + 146^2}} \approx \frac{3332}{3540} \approx 0.941$$

where $\theta$ is the angle between the vectors of *Mary* and *girl*; and $\phi$ is the angle between the vectors of *Mary* and *boy*.

In the current example of a naïve vector space, *John* is also closer to *girl* than to *boy*, which is counter-intuitive. This might be because of the small number of dimensions used, the poor selection of the context words, or the usage of raw co-occurrence numbers.

## 2.4 Representation of phrases and sentences

Both local and global distributional approaches have advantages over the formal approach in their ability to capture lexical semantics and degrees of similarity. However, their success at extending this to the sentence level, and to more complex semantic phenomena, depends on their applicability within compositional models.

### 2.4.1 The principle of compositionality

Formal approaches to the semantics of natural language have built upon the classical idea of compositionality which states that the meaning of a sentence is a function of its parts (Janssen 2001). Syntactic rules define how constituents are recursively combined to form other constituents until the whole sentence is covered. Translation rules define how semantic representations of the constituents are combined to get a semantic representation of the whole.

In compositional type-logical approaches, predicate-argument structures representing phrases and sentences are built from their constituent parts by general operations such as beta-reduction within the lambda calculus (Montague 1970): for example, given a semantic representation of *John* as *John*$'$, *loves* as $\lambda y.\lambda x.\textit{loves}'(x, y)$ and *Mary* as *Mary*$'$, the semantic representation of the sentence *John loves Mary* can be constructed as

$$\lambda y.\lambda x.\textit{loves}'(x, y)(\textit{mary}')(\textit{john}') = \textit{loves}'(\textit{john}', \textit{mary}')$$

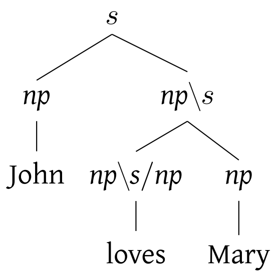


**Figure 2.1:** A syntactic tree for *John loves Mary*. The lexicon assigns categories to words: *John* is *np*, loves is $np\backslash s/np$ and Mary is *np*. Backward and forward composition rules derive the syntactic tree.

Categorical grammars are widely used to obtain the syntactic structure of a sentence. Given a set of basic categories `ATOM`, for example $\{n, s, np\}$, complex categories `CAT\CAT` and `CAT/CAT` can be constructed, where `CAT` is either an element of `ATOM` or a complex category. So the category of a transitive verb is `np\s/np`. Intuitively, we want to express that to obtain a sentence with a transitive verb there must be two noun phrases, one before and another after the verb.

Parsing is done by composing categories together according to two rules:

1. **Backward application**: If $\alpha$ is a string of category $A$ and $\beta$ is a string of category $A\backslash B$, then $\alpha\beta$ is of category $B$.

2. **Forward application**: If $\alpha$ is a string of category $A$ and $\beta$ is a string of category $B/A$, then $\beta\alpha$ is of category $B$.

Figure 2.1 illustrates the parse tree for *John loves Mary* obtained using the category composition rules.

The last step is to map syntactic categories with semantic terms. Again, there are base types ($e$ for entities and $t$ for sentences) and complex types of the form $(a \rightarrow b)$ where $a$ and $b$ are types. The mapping between syntactic categories and semantic types is defined as a function *type*:

$$\begin{aligned}
&\mathit{type}(np) = e\\
&\mathit{type}(s) = t\\
&\mathit{type}(A/B) = (\mathit{type}(B) \rightarrow \mathit{type}(A))\\
&\mathit{type}(B\backslash A) = (\mathit{type}(B) \rightarrow \mathit{type}(A))
\end{aligned}$$

Syntactic backward and forward application corresponds to functional application. Figure 2.2 shows the final parse tree.

Given a suitable pairing between a syntactic grammar, semantic representations and corresponding general combinatory operators, this can produce structured sentential representations with a broad coverage and good generalisability (Bos 2008). This logical approach is extremely powerful because it can capture complex aspects of meaning such as quantifiers

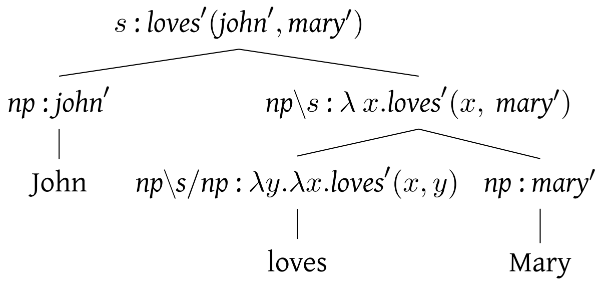

$s : loves'(john', mary')$
$np : john'$
$np\backslash s : \lambda\, x.loves'(x,\ mary')$
John
$np\backslash s/np : \lambda y.\lambda x.loves'(x, y)$
$np : mary'$
loves
Mary

**Figure 2.2:** The final parse tree

and their interactions (Copestake et al. 2005), and enables inference using well studied and developed logical methods (Bos and Gabsdil 2000).

### 2.4.2 Compositional distributional semantics

Methods based on the distributional hypothesis have been recently applied to many tasks, but mostly at the word level, for instance, word sense disambiguation (Zhitomirsky-Geffet and Dagan 2009) and lexical substitution (Thater et al. 2010). They exploit the notion of similarity which correlates with the angle between word vectors (Turney and Pantel 2010).

*Compositional* distributional semantics goes beyond word level and models the meaning of phrases or sentences based on their parts. Mitchell and Lapata (2008) perform composition of word vectors using vector addition and multiplication operations. The limitation of this approach is the operator commutativity, which ignores the argument order, and thus the syntactic/semantic function of the words (as a consequence for English, the word order is ignored). As a result, *John loves Mary* and *Mary loves John* get identical representations.

#### Notation

Before introducing the compositional operators, the used notation is explained here.

Vectors of words are written with an arrow, for example $\overrightarrow{mary} = [0, 10, 22]$. The arrow above a word indicates that the representation of it is a vector. The vector values are either raw co-occurrence counts obtained from a corpus, as in the examples below, or their quantified counterparts.

Point-wise addition, written as $+$, and point-wise multiplication, written as $\odot$, are the two basic compositional operators. As the output of these operators is a vector, the vector that represents a sentence is written as the sentence itself with an arrow on top and the operator used in the bottom right: for example, $\overrightarrow{\textit{John loves Mary}}_{\text{addition}}$ and $\overrightarrow{\textit{John loves Mary}}_{\text{multiplication}}$ for addition and multiplication respectively.

Other operators require a verb to be represented as a matrix. The matrix can be obtained in various ways, in this work two procedures are used.

The Kronecker compositional operator (see below) uses the Kronecker product, written as $\otimes$, which is a generalization of the outer product:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix} \tag{2.1}$$

where **A** is a $m \times n$ matrix and **B** is $p \times q$ matrix. In this work, vectors are used to produce verb matrices, making the actual computation simpler (for the vectors $\vec{a}$ and $\vec{b}$ of length $n$):

$$\vec{a} \otimes \vec{b} = \begin{bmatrix} a_1 b_1 & \cdots & a_1 b_n \\ \vdots & \ddots & \vdots \\ a_n b_1 & \cdots & a_n b_n \end{bmatrix} \tag{2.2}$$

Verb matrices used by the Kronecker compositional operator are written as $\widetilde{Verb} = \overrightarrow{Verb} \otimes \overrightarrow{Verb}$. The tilde indicates that it is a matrix that, in turn, is the Kronecker product of the verb vector with itself.

As the output of the Kronecker compositional operator is a matrix, the representation of a sentence obtained using it is written with a line instead of an arrow, for instance, $\overline{\textit{John loves Mary}}_{\text{Kronecker}}$, the operator is identified in the bottom right.

The second kind of verb matrices is written as $\overline{Verb}$ and obtained from a corpus by considering the Kronecker products of the subject-object pairs that occur with the verb, see below.

Finally, $^{\mathsf{T}}$ transposes a matrix. The transpose of the matrix **A** is written as $\mathbf{A}^{\mathsf{T}}$ and $\mathbf{A}_{ij} = \mathbf{A}^{\mathsf{T}}_{ji}$.

## Compositional operators

Consider that *John*, *Mary* and *loves* are represented as vectors $\overrightarrow{john} = [4, 60, 59]$, $\overrightarrow{mary} = [0, 10, 22]$ and $\overrightarrow{loves} = [10, 100, 20]$, respectively. Then the vector of the sentence *John loves Mary* using addition is:

$$\begin{aligned} \overrightarrow{\textit{John loves Mary}}_{\text{addition}} &= \overrightarrow{john} + \overrightarrow{loves} + \overrightarrow{mary} \\ &= [4, 60, 59] + [10, 100, 20] + [0, 10, 22] \\ &= [14, 170, 101] \end{aligned}$$

It is similar for multiplication, where element-wise multiplication is used instead of addition:

$$\begin{aligned}
\overrightarrow{\textit{John loves Mary}}_{\text{multiplication}} &= \overrightarrow{\textit{john}} \odot \overrightarrow{\textit{loves}} \odot \overrightarrow{\textit{mary}} \\
&= [4, 60, 59] \odot [10, 100, 20] \odot [0, 10, 22] \\
&= [0, 60\,000, 25\,960]
\end{aligned}$$

To capture word order and the syntactic structure of a sentence, various approaches have been proposed. Grefenstette and Sadrzadeh (2011a) use non-commutative linear algebra operators as proposed in the theoretical work of Coecke et al. (2010). There, the functional applications of semantic terms are replaced with tensors (Bourbaki 1998).

Kronecker (Grefenstette and Sadrzadeh 2011b) is a non-commutative operator. It represents the verb of a phrase as a matrix and the subject and the object as vectors. The verb matrix is defined as the Kronecker product—which gives the name to the compositional operator—of the vector of a verb with itself:

$$\widetilde{\textit{Verb}} = \overrightarrow{\textit{Verb}} \otimes \overrightarrow{\textit{Verb}} \tag{2.3}$$

The representation[3] of a subject-verb-object phrase is computed as:

$$\overline{\textit{Sbj Verb Obj}} = \widetilde{\textit{Verb}} \odot (\overrightarrow{\textit{Sbj}} \otimes \overrightarrow{\textit{Obj}}) \tag{2.4}$$

The derivation of the representation of *John loves Mary* using Kronecker is:

$$\begin{aligned}
\overline{\textit{John loves Mary}}_{\text{Kronecker}} &= \widetilde{\textit{loves}} \odot (\overrightarrow{\textit{john}} \otimes \overrightarrow{\textit{mary}}) \\
&= (\overrightarrow{\textit{loves}} \otimes \overrightarrow{\textit{loves}}) \odot (\overrightarrow{\textit{john}} \otimes \overrightarrow{\textit{mary}}) \\
&= ([10, 100, 20] \otimes [10, 100, 20]) \odot ([4, 60, 59] \otimes [0, 10, 22]) \\
&= \begin{bmatrix} 100 & 1\,000 & 200 \\ 1\,000 & 10\,000 & 2\,000 \\ 200 & 2\,000 & 400 \end{bmatrix} \odot \begin{bmatrix} 0 & 40 & 88 \\ 0 & 600 & 1\,320 \\ 0 & 590 & 1\,298 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 40\,000 & 17\,600 \\ 0 & 6\,000\,000 & 2\,640\,000 \\ 0 & 1\,180\,000 & 519\,200 \end{bmatrix}
\end{aligned}$$

For other non-commutative operators presented below, a transitive verb is represented by the matrix $\overline{\textit{Verb}}$, which is obtained from a corpus using the formula (Grefenstette and Sadrzadeh 2011a):

$$\overline{\textit{Verb}} = \sum_i \overrightarrow{s_i} \otimes \overrightarrow{o_i} \tag{2.5}$$

where $\overrightarrow{s_i}$ and $\overrightarrow{o_i}$ are the subject-object pairs of the verb found in the source corpus.

[3] The operators that include Kronecker product produce matrices, not vectors, however computing similarity between matrices is possible by collapsing the rows of the matrix into one dimension by concatenating them together.

This encodes into the verb matrix all the contextual information of the subjects and objects of the verb. The way it is constructed (by pairing subjects and objects of the verb) is the same way a verb predicate is built in formal semantics, that is by pairing the subjects and objects. But here, instead of packing the pairs into a set, one develops a matrix from them, at the same time, each pair gets a real number value assigned to it.

To continue the example, imagine that the verb *loves* was seen three times in the corpus: *John loves Mary*, *Peter likes coffee* and *Mary likes hiking*. The matrix $\overline{loves}$ is computed as:

$$\overline{loves} = (\overrightarrow{john} \otimes \overrightarrow{mary}) + (\overrightarrow{peter} \otimes \overrightarrow{coffee}) + (\overrightarrow{mary} \otimes \overrightarrow{hiking}) = \begin{bmatrix} 120 & 1\,390 & 658 \\ 1\,360 & 13\,670 & 2\,874 \\ 390 & 3\,930 & 846 \end{bmatrix}$$

Relational compositional operator is defined as (Grefenstette and Sadrzadeh 2011a):

$$\overline{Verb} \odot (\overrightarrow{Sbj} \otimes \overrightarrow{Obj}) \tag{2.6}$$

Informally, it identifies the interaction of the features of the subject and the object by the Kronecker product $\overrightarrow{Sbj} \otimes \overrightarrow{Obj}$ that are later filtered by verb using point-wise multiplication (Grefenstette and Sadrzadeh 2011a). The filtering by verb is needed because the subject and the object can belong to several relations.

In the example, the meaning is computed similarly to Kronecker, but a different verb matrix is used:

$$\begin{aligned} \overline{John\ loves\ Mary}_{\text{relational}} &= \overline{loves} \odot (\overrightarrow{john} \otimes \overrightarrow{mary}) \\ &= \begin{bmatrix} 120 & 1\,390 & 658 \\ 1\,360 & 13\,670 & 2\,874 \\ 390 & 3\,930 & 846 \end{bmatrix} \odot \begin{bmatrix} 0 & 40 & 88 \\ 0 & 600 & 1\,320 \\ 0 & 590 & 1\,298 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 55\,600 & 57\,904 \\ 0 & 8\,202\,000 & 3\,793\,680 \\ 0 & 2\,318\,700 & 1\,098\,108 \end{bmatrix} \end{aligned}$$

The categorical framework of Coecke et al. (2010) requires the meaning of a transitive verb to be a tensor of third order (it must be a cube), but the relational method of Grefenstette and Sadrzadeh (2011a) constructs verb meanings as tensors of second order (they are matrices). To apply the categorical framework, Kartsaklis et al. (2012) embed the second order verb matrix into a third order space by a map $\sigma : N^2 \to N^3$. This map is one of the maps of the Frobenius algebra over vector space N.

The Frobenius operation $\sigma$ turns a matrix into a cube by filling the extra dimension of the cube and gives some options on how the mapping can be implemented. The operators copy-subject and copy-object (Kartsaklis et al. 2012) implement this idea. The cube is formed using copy-subject (copy-object), when one applies cube contraction on it with the object (the subject) and then matrix multiplication with the subject (the object).

Copy-object diagonally places the object dimension into a cube, producing a closed formformula:

$$\overrightarrow{Sbj} \odot (\overline{Verb} \times \overrightarrow{Obj}) \tag{2.7}$$

where × is matrix multiplication. The vector representation of *John loves Mary* using copy-object is:

$$\begin{aligned}
\overline{John\ loves\ Mary}_{\text{copy-object}} &= \overrightarrow{john} \odot (\overline{loves} \times \overrightarrow{mary}) \\
&= [4, 60, 59] \odot \left( \begin{bmatrix} 120 & 1\,390 & 658 \\ 1\,360 & 13\,670 & 2\,874 \\ 390 & 3\,930 & 846 \end{bmatrix} \times [0, 10, 22] \right) \\
&= [4, 60, 59] \odot [28\,376, 199\,928, 57\,912] \\
&= [113\,504, 11\,995\,680, 3\,416\,808]
\end{aligned}$$

Similarly, copy-subject diagonally places the subject dimension into a cube, producing a formula:

$$\overrightarrow{Obj} \odot (\overrightarrow{Verb}^{\mathsf{T}} \times \overrightarrow{Sbj}) \tag{2.8}$$

Again, in the example:

$$\begin{aligned}
\overline{John\ loves\ Mary}_{\text{copy-subject}} &= \overrightarrow{mary} \odot (\overline{loves}^{\mathsf{T}} \times \overrightarrow{john}) \\
&= [0, 10, 22] \odot \left( \begin{bmatrix} 120 & 1\,360 & 390 \\ 1\,390 & 13\,670 & 3\,930 \\ 658 & 2\,874 & 846 \end{bmatrix} \times [4, 60, 59] \right) \\
&= [0, 10, 22] \odot [105\,090, 1\,057\,630, 224\,986] \\
&= [0, 10\,576\,300, 4\,949\,692]
\end{aligned}$$

The series of contraction and multiplication simplifies to these closed forms, due to the special shape of the cube (that is created by copy-subject or copy-object). Note that the closed-form formulas do not include third order tensors.

Kartsaklis and Sadrzadeh (2014) observe that copy-object and copy-subject address a partial interaction of the verb with its arguments and propose a family of three *Frobenius operators* that combine copy-object and copy-subject together.

Frobenius addition uses addition to combine the results:

$$\big(\overrightarrow{Sbj} \odot (\overline{Verb} \times \overrightarrow{Obj})\big) + \big(\overrightarrow{Obj} \odot (\overline{Verb}^{\mathsf{T}} \times \overrightarrow{Sbj})\big) \tag{2.9}$$

So the computation of *John loves Mary* is:

$$\begin{aligned}\overline{\textit{John loves Mary}}_{\text{Frobenius add}} &= \overline{\textit{John loves Mary}}_{\text{copy-object}} + \overline{\textit{John loves Mary}}_{\text{copy-subject}} \\ &= [113\,504, 11\,995\,680, 3\,416\,808] + [0, 10\,576\,300, 4\,949\,692] \\ &= [113\,504, 22\,571\,980, 8\,366\,500]\end{aligned}$$

Element-wise vector multiplication is used by Frobenius multiplication:

$$(\overrightarrow{Sbj} \odot (\overline{Verb} \times \overrightarrow{Obj})) \odot (\overrightarrow{Obj} \odot (\overline{Verb}^{\mathsf{T}} \times \overrightarrow{Sbj})) \tag{2.10}$$

Similarly, the computation of *John loves Mary* becomes:

$$\begin{aligned}\overline{\textit{John loves Mary}}_{\text{Frobenius mult}} &= \overline{\textit{John loves Mary}}_{\text{copy-object}} \odot \overline{\textit{John loves Mary}}_{\text{copy-subject}} \\ &= [113\,504, 11\,995\,680, 3\,416\,808] \odot [0, 10\,576\,300, 4\,949\,692] \\ &= [0, 126\,869\,910\,384\,000, 16\,912\,147\,223\,136]\end{aligned}$$

Finally, Kronecker product is used in Frobenius outer:

$$(\overrightarrow{Sbj} \odot (\overline{Verb} \times \overrightarrow{Obj})) \otimes (\overrightarrow{Obj} \odot (\overline{Verb}^{\mathsf{T}} \times \overrightarrow{Sbj})) \tag{2.11}$$

Making the computation of *John loves Mary* look like:

$$\begin{aligned}\overline{\textit{John loves Mary}}_{\text{Frobenius outer}} &= \overline{\textit{John loves Mary}}_{\text{copy-object}} \otimes \overline{\textit{John loves Mary}}_{\text{copy-subject}} \\ &= [113\,504, 11\,995\,680, 3\,416\,808] \otimes [0, 10\,576\,300, 4\,949\,692] \\ &= \begin{bmatrix} 0 & 1\,200\,452\,355\,200 & 56\,1809\,840\,768 \\ 0 & 126\,869\,910\,384\,000 & 59\,374\,921\,330\,560 \\ 0 & 36\,137\,186\,450\,400 & 16\,912\,147\,223\,136 \end{bmatrix}\end{aligned}$$

### 2.4.3 Similarity of heads in context

A noun phrase can be similar to a noun, as in *female lion* and *lioness*, and to other noun phrases as in *yellow car* and *cheap taxi*. The same similarity principle can be applied to phrases as well as to words. In this case, similarity is measured in *context*,[4] and most methods of calculating similarity of phrases still rely on comparisons of the phrases' head words, which meanings are modified by the arguments they appear with (Kintsch 2001).

Mitchell and Lapata (2008; 2010) use element-wise addition and multiplication to model argument interaction. Baroni and Zamparelli (2010) represent adjectives as matrices that modify nouns (vectors) using matrix multiplication.

[4] The similarity between the head words in the context of their phrases should not be confused with contextualised similarity. The similarity between heads specifies what is being compared. Contextualised similarity is a requirement on the similarity relation.

Dinu and Lapata (2010) model word meaning as a distribution over senses; a context feature (other word in the context) directly modulates word's sense distribution using conditional probability. Thater et al. (2011) contextualise vectors by assigning higher weight to features that correspond or are distributionally similar to the context words.

With verbs, similarity in context can be applied to compare a transitive verb with an intransitive verb. For example, *cycle* is similar to *ride a bicycle*. Here, we see that *a bicycle* disambiguates the verb *ride* making the phrase similar to the verb *cycle*. The connection between measuring similarity of a phrase and disambiguation has been noted in Kartsaklis et al. (2013).

Sentential similarity might be treated as the similarity of the heads in their contexts. That is, the similarity between *sees* and *notices* in *John **sees** Mary* and *John **notices** a woman*. This approach abstracts away the grammatical difference between the sentences and concentrates on their semantics.

## 2.5 Evaluation

The difference in occurrence between two words quantifies the difference in their meaning (Harris 1954) and there is a procedure to measure the difference in the meanings of two sentences (Coecke et al. 2010). The final necessary part is the evaluation methodology to test this approach.

Because it is difficult to perform extrinsic evaluation (also called evaluation in use) to measure the performance of a similarity component in a pipeline of a complete natural language processing system (for example, a dialog system), intrinsic datasets that focus on similarity are popular among computational linguists.

Apart from a pragmatic attempt to alleviate the problems of evaluating similarity components, these datasets serve as an empirical test of the hypotheses of Firth and Harris. They bring together our understanding of the human mind, language and technology. The following sections introduce the main datasets used.

### 2.5.1 Word similarity

Rubenstein and Goodenough (1965) performed an empirical study on the relationship between the similarity in word meaning and similarity of contexts they appear in. They build a list of 65 word pairs that range from highly synonymous to semantically unrelated.

To obtain the gold standard similarity judgements, the human subjects were given a shuffled deck of cards. Each card contained a word pair from the list. They were asked to sort the cards by similarity, so that the most similar items appeared in the top of the deck. In addition to ranking, human subjects were asked to give similarity scores ranging from 4.0 to 0.0, where the higher value indicates the higher similarity level.

To test the Distributional hypothesis, 100 sentences for each distinct word in the list were written by 50 participants who did not provide similarity judgements to be used as the contexts. The words from the list had to be used as nouns and the sentences had to be at least 10 words long.

To estimate similarity, the overlap was calculated over tokens and types. The token condition takes into account the appearance frequencies of context words, while the type condition only considers the word types. So if the word *table* appeared 10 times as a context of a word of interest, in the case of the token condition all 10 occurrences are considered, in the case of the type condition only the fact that it appeared as a context is recorded.

Formally, the overlap $M_x$ over condition $x$ between the word context of words $A$ and $B$ was calculated as:

$$M_x = \frac{N(A_x B_x)}{\min(N(A_x), N(B_x))}$$

where $N(A_x B_x)$ is the number of context words shared between $A$ and $B$ under condition $x$. $N(A_x)$ and $N(B_x)$ are the number of context words under condition $x$ for the words $A$ and $B$ respectively.

They found that the more similar words in meaning are the more context they share for both token and type conditions. Moreover, the relationship is strongest for the highly synonymous pairs, namely the pairs with similarity greater than 3.0.

Tversky and Hutchinson (1986) studied the similarity relation from the psychological perspective. They analysed the similarity judgements between the entries by their geometric structure. The geometric approach represents the entries as points in a multidimensional space so that the distance between them reflects similarity.

They examined 100 datasets to identify common geometric patterns in human similarity judgements. The datasets contained entries that belonged to a single category such as *verbs of judging* (Fillenbaum and Rapoport 1974) or *animal terms* (Henley 1969). The reason for category oriented similarity studies is that "stimuli can only be compared in so far as they have already been categorised as identical, alike, or equivalent at some higher level of abstraction" (Turner et al. 1987).

They observed that if a category contains a superordinate, similarity judgements arrange category members around it. For example, similarity judgements given by humans arrange fruit names around the word *fruit* in such a way that it is their nearest neighbour, making *fruit* the *focal point* of the category of *fruits*.

An important consequence is that high centrality values cannot be achieved in a space with dimensionality of two or three, because the dimensionality sets the upper bound on the number of points that can share the nearest neighbour.

Finkelstein et al. (2002) proposed a context oriented information retrieval framework. The idea that the meaning of the word *jaguar* is dependent on the context the search is performed. It might be a car, if the query comes from an automotive website, or it might mean an animal if

| Model | WS353 | MEN | SimLex-999 |
|---|---|---|---|
| Finkelstein et al. (2002) | 0.55 | | |
| Bruni et al. (2012) | 0.75 | 0.76 | |
| Kiela and Clark (2014) | 0.58 | 0.71 | |
| Baroni et al. (2014b) | | | |
| Distributional model | 0.62 | 0.72 | |
| Neural word embeddings | 0.73 | 0.80 | |
| Hill et al. (2015) | | | |
| Distributional model | 0.42 | 0.44 | 0.19 |
| Neural word embeddings | 0.44 | 0.43 | 0.28 |
| Levy et al. (2015) | | | |
| Distributional model | 0.75, 0.70 | 0.75 | 0.39 |
| Neural word embeddings | 0.79, 0.69 | 0.77 | 0.44 |
| State of the art | 0.81 | 0.80 | 0.76 |
| Upper bound | | 0.84 | 0.78 |

**Table 2.2:** Model performance on various datasets. For Bruni et al. (2012) the model with the highest average score is shown (TunedFL, Window20). Levy et al. (2015) report two results on WS353: on the subset that contains similar items (the first number) and the subset that contains related items (the second number). The state of the art are Halawi et al. (2012) for WS353, Baroni et al. (2014b) for MEN and Recski et al. (2016), which is a not pure distributional model, for SimLex-999.

it comes from a website about nature. To evaluate their semantic component they developed a dataset WS353 that consists of 353 diverse noun pairs along with their relatedness scores on a scale from 0 (totally unrelated) to 10 (very much related or identical). The combination of a vector-based method and a WordNet-based method achieved correlation of 0.55. The dataset they proposed is widely used to evaluate algorithms that estimate semantic similarity.

Bruni et al. (2012) introduced the MEN dataset to test a multimodal semantic space (the model used textual and visual features). The new dataset contains the words that appear in labels of the ESP-Game[5] and MIRFLICKR-1M[6] image collections. Compared to WS353 it is sufficiently large to be split to the development part (2 000 pairs) and to the test part (1 000 pairs) for evaluation. The dataset contain highly similar items (*cathedral*, *church*, 0.94) and also terms that stand in a broader semantic relationship, such as whole-part (*flower*, *petal*, 0.92). The scores are relatedness scores on a scale from 0.0 to 1.0.

The Spearman correlation of the judgements given by two authors of the paper for all 3 000 pairs is 0.68. The correlation with the average of their scores with the dataset scores is 0.84, which can be taken as the upper bound. The best presented results are 0.75 for WS353 and 0.78 for MEN, note that two different models produce them.

Hill et al. (2015) presented SimLex-999, a gold standard resource for evaluating distributional semantic models. In contrast to WS353 and MEN, it focuses on similarity rather than relatedness. The words *coffee* and *cup* are related but not similar. The dataset consists of 666 noun-pairs, 222 verb-pairs and 111 adjective-pairs. 500 residents of the USA were recruited to collect

[5] http://www.cs.cmu.edu/~biglou/resources/

[6] http://press.liacs.nl/mirflickr/

human judgements. The average pairwise Spearman correlation between two participants is 0.67, however the average correlation of a human rater with the average of all other raters is 0.78.

They tested several models on WS353, MEN and WordSim-999. They showed that existing models achieved lower correlation on SimLex-999 than on WS353 and MEN suggesting that the new dataset required development of methods that are strictly focused on similarity. The best reported results in that study are 0.44 on WS353, 0.48 on MEN and 0.28 on SimLex-999, these are different models, but all of them are trained on the 150 million word RCV1 Corpus (Lewis et al. 2004). Neural models trained on a larger corpus (Wikipedia) yield higher results as shown by Levy et al. (2015) and Baroni et al. (2014b).

Table 2.2 presents the key models and the results they achieved on WS353, MEN and SimLex-999.

### 2.5.2 Parameter selection for lexical models

The variance of results on word similarity tasks can be explained by the differences in algorithms or by difference in model parameters, corpus used and other factors. For example, Baroni et al. (2014b) uses a corpus of 2.8 billion tokens and outperforms Hill et al. (2015), who uses a corpus of 0.15 billion tokens, on MEN by 0.28 points with a distributional model and by 0.37 with neural word embeddings.

Bullinaria and Levy (2007) presented a thorough study of parameters of distributional models. They tested various vector space parameters and similarity measures.

Instead of raw co-occurrence counts they used the conditional probability of a context word appearing close to a target word $P(c|t)$. In addition to that, they used *point-wise mutual information* $\mathrm{PMI(c,t)} = \log\left(\frac{P(c,t)}{P(c)P(t)}\right)$ and positive PMI, which nullifies negative values, and probability ratio $\frac{P(c|t)}{P(c)}$. They varied the vector space dimensionality from 1 to 100 000 dimensions. Cosine, Euclidean and City Block geometric measures were used to estimate similarity. They used the 87.9 million word British National Corpus (BNC) and performed an exhaustive search across the parameter combinations.

A vector space with vector components computed with positive PMI and cosine similarity measure performed best in their experiments. They also showed that performance depends on the size of the corpus used: the larger the corpus, the better the results. The context window of size one produced the best results. The model performance peaked when 1 000 dimensional vector space was used and dropped as dimensionality increased afterwards.

Kiela and Clark (2014) performed a systematic study of distributional models on various datasets including WS353 and MEN. They confirmed findings of Bullinaria and Levy (2007) that larger corpora lead to better performance. They suggested using ukWaC (2 billion tokens) over the BNC with a small context window of size less than 5 from each side. Positive PMI was the best performed measure in combination with the *correlation similarity* measure, which is the mean-adjusted cosine similarity. The dimensionality of 50 000 appeared to be optimal.

Kiela and Clark (2014) advocated incremental parameter tuning, reasoning that compositional tasks need a vector space model that is good for lexical tasks. Practically, this means that initially a good lexical model is identified and only then it is used to find a well performing compositional operator. Bullinaria and Levy (2012) followed the same reasoning: first good sparse models are identified, and then dimensionality reduction methods are compared to find the best space with reduced dimensionality.

Lapesa and Evert (2013) argue against incremental tuning because it does not capture parameter interaction. For example, a compositional operator might benefit from a specific weighting scheme, which is not necessarily the best in predicting word similarity. Their goal was to contrast rank-based prediction of semantic priming with distance-based prediction. Because they were introducing rank-based prediction, they could not reuse recommendations of parameters that are derived from distance-based similarity experiments. In their study, they covered a broad range of parameters of distributional models aiming at identification of parameter configurations that achieve good performance in predicting semantic priming.

They used a linear regression model to determine the importance of individual parameters and their combinations. The parameters of a distributional model—such as the weighting scheme, vector space dimensionality and similarity metric—were considered predictors of its performance: the parameters of a distributional model were independent parameters of a linear model and the performance of the distributional model was a dependent variable. Analysis of variance was used to determine the most important parameters and their interactions.

They showed that a statistical association measures—such as t-score or z-score (Evert 2005)—with the combination of ranked-based prediction yielded better results over cosine similarity. Would they have performed only iterative tuning, they would test ranked-based prediction only with the PMI weighting, which underperformed in their experiments.

In a successive study that included word similarity, Lapesa and Evert (2014) showed that the combination of the similarity score together with the weighting scheme was the most important parameter combination, supporting the findings of Bullinaria and Levy (2007) and Kiela and Clark (2014), who suggested to use PMI together with cosine similarity.

Baroni et al. (2014b) and Levy et al. (2015) performed systematic parameter studies on distributional and neural models. Levy et al. (2015) evaluated models on all three datasets (WS353, MEN and SimLex-999, see Table 2.2). Notably, their best distributional model outperformed the distributional models presented in Hill et al. (2015), Kiela and Clark (2014) and Baroni et al. (2014b) and to the best of our knowledge is the state of the art of distributional models.

### 2.5.3 Statistical significance testing

Rastogi et al. (2015) proposed and Faruqui et al. (2016) expressed the importance of a method

| Dataset | $\sigma_{0.05}^{0.9}$ |
|---|---|
| SimLex-999 | 0.023 |
| MEN | 0.013 |
| KS14 | 0.07 |
| GS11 | 0.05 |
| PhraseRel | 0.43 |

**Table 2.3:** Minimal required difference for significance (MRDS) numbers for the datasets used in this work. Values for SimLex-999 and MEN are taken from Rastogi et al. (2015). KS14, GS11 and PhraseRel are described later in this work.

for significance testing motivated by the fact that the researchers do not report measures of significance of the difference between the Spearman correlations. Their method is based on finding a minimal required difference for significance (MRDS).

Consider two lists of ratings over the same dataset: $A$ and $B$ produced by two competing models together with a list of gold ratings $T$. Let $r_{AT}$, $r_{BT}$, and $r_{AB}$ be the Spearman correlations between $A$:$T$, $B$:$T$ and $A$:$B$ respectively. Let $\hat{r}_{AT}$, $\hat{r}_{BT}$ and $\hat{r}_{AB}$ be their empirical estimates and assume without loss of generality that $\hat{r}_{BT} > \hat{r}_{AT}$.

The MDRS of a dataset $\sigma_{p_0}^{r}$ is defined such that it satisfies the following:

$$(r_{AB} < r) \wedge (|\hat{r}_{BT} - \hat{r}_{AT}| < \sigma_{p_0}^{r} \Rightarrow \textit{pval} > p_0 \tag{2.12}$$

In the constraint above, *pval* is the probability of the test statistic under the null hypothesis that $r_{AT} = r_{BT}$ found using Staiger's test (Steiger 1980).

The constraint ensures that the $p$value of the null hypothesis will be greater than $p_0$ given that the correlation between the methods is less than $r$ and the difference between the correlations of the competing models to the gold standard is less then $\sigma_{p_0}^{r}$.

The value of $r$ specifies the upper bound on the agreement of the ratings produced by the competing models, for example $r = 0.9$.

The second part of the predicate ensures that the null hypothesis is more likely than $p_0$ given that the difference between the correlations of the models to the gold standard is less than $\sigma_{p_0}^{r}$.

Once a reasonable value of $r$ is chosen (for example 0.9), $\sigma_{p_0}^{r}$ can be found following this procedure. Let stest be Staiger's test predicate which satisfies the following ($n$ being the size of the dataset):

$$\text{stest-p}(\hat{r}_{AT}, \hat{r}_{BT}, r_{AB}, p_0, n) \Rightarrow \textit{pval} < p_0 \tag{2.13}$$

Now it is possible to search for $\sigma_{p_0}^{r}$ by solving:

$$\sigma_{p_0}^{r} = \min\{\sigma | \forall_{0<r'<1} \text{ stest-p}(r', \min(r' + \sigma, 1), r, p_0, n)\} \tag{2.14}$$

The $\sigma_{0.05}^{0.9}$ values for the lexical datasets and phrasal datasets (see next section and Chapter 5) are listed in Table 2.3.

### 2.5.4 Disambiguation of verbs in context

The transitive verb disambiguation dataset (GS11) described in Grefenstette and Sadrzadeh (2011a;b) consists of ambiguous transitive verbs together with their arguments, landmark verbs (which identify one of the verb senses) and human judgements (which specify the similarity to the landmarks of the disambiguated sense of the verb in the given context). This is similar to the intransitive dataset described in Mitchell and Lapata (2008).

Consider the sentence *system meets specification*. *Meets* is an ambiguous transitive verb, and *system* and *specification* are its arguments. Possible landmarks for *meet* are *satisfy* and *visit*. For this sentence, the human judgements show that the disambiguated verb meaning is similar to the landmark *satisfy*, and less similar to *visit*.

The task is to estimate the similarity of the sense of a verb in a context with a given landmark. To estimate similarity, the verb is composed with its arguments, it is done the same for the landmark and the arguments, and the similarity of the two phrase vectors is computed. To evaluate performance, the human judgements are averaged for the same verb, argument and landmark entries, and these average values are used to calculate the correlation. Grefenstette and Sadrzadeh (2011b) achieve the correlation of 0.28 with the composition based on Kronecker.

### 2.5.5 Sentence similarity

The transitive sentence similarity dataset (KS14) described in Kartsaklis and Sadrzadeh (2014), Kartsaklis et al. (2013) consists of transitive sentence pairs and human similarity judgements. The task is to estimate similarity between two sentences. The evaluation is the same as in the disambiguation task (Section 2.5.4). They achieve the correlation of 0.41 with additive composition.

### 2.5.6 Parameter selection for compositional models

In general, there is no systematic study of parameters of compositional models similar to lexical studies discussed in Section 2.5.2. First of all, composition brings another type of parameters, so not only lexical representations have to be optimised, but also an optimal way of composition has to be found.

Early compositional studies focused on the compositional operators, using lexical representations that are good for lexical tasks. Because parameters were selected iteratively, the fact that one operator outperformed other could be due to the specificity of lexical representations.

Nevertheless, experiments based on iterative parameter selection gave positive results, see, for example, the papers that introduced the compositional datasets described above Grefenstette and Sadrzadeh (2011b), Kartsaklis et al. (2013).

Blacoe and Lapata (2012) is—to the best of our knowledge—the first paper that, apart from identifying the best compositional operator, also explicitly contrasted two kinds of lexical representations: shallow that are based on the co-occurrence of words and embeddings that are learned by a neural language model (Bengio et al. 2006, Collobert and Weston 2008). They concluded that shallow approaches are as good as their computationally more intensive counterparts based on language models in phrase similarity and paraphrase detection tasks. They also identified the importance of the combination of lexical representation and the compositional method.

Milajevs et al. (2014), the work that set the ground for this thesis, also contrasts count and predict (Baroni et al. 2014b) models in compositional setting. However, in contrast to Blacoe and Lapata (2012) experiments, prediction-based models showed robust performance yielding better results on a number of tasks and were recommended for usage in compositional experiments. The work consecutive to Milajevs et al. (2014) (consequently or coincidentally) focuses on predictive lexical representations.

The CBOW model of local lexical representations (Mikolov et al. 2013a;b;c) implicitly assumes additive interaction between context words to predict the target word. However, categorical compositional operators (Section 2.4.2) make use of multiplication as a part of composition. This inconsistency was noticed in Kim et al. (2015b). They changed the objective function to include multiplication:

$$\frac{1}{T}\sum_{t=1}^{T}\prod_{-c\leqslant j\leqslant c, j\neq 0}\log P(w_t|w_{t+j}) \tag{2.15}$$

This change improved results of categorical compositional methods on the sentence similarity dataset (Section 2.5.5), but not on the verb disambiguation task (Section 2.5.4).

In the categorical framework, the result of composition depends on how the tensors for transitive verbs are constructed. The most straightforward approach is to represent transitive verbs as matrices (saving a dimension) as it is shown in (2.5) on page 26. However, the number of elements in these matrices is still high and equal to the square of the number of elements in vectors for nouns. To cope with this problem various solutions were tried: linear-regression of the full tensors (Grefenstette et al. 2013), the combination of regression with a plausibility space (Polajnar and Clark 2014) and low-rank tensor approximation (Fried et al. 2015). While these methods not necessarily lead to the highest results, they considerably reduce the amount of data associated with lexical entries.

Finally, Hashimoto and Tsuruoka (2015), Hashimoto et al. (2014) proposed a tensor factorisation method that directly models the interaction between predicates and their arguments. The intuition is that the arguments should be able to disambiguate the meaning of their predicate.

### 2.5.7 Extrinsic evaluation

It is common to evaluate word meaning representations and compositional models in a pipeline, for example dialogue act tagging (Stolcke et al. 2000) or paraphrase detection (Dolan and Brockett 2005). However, these datasets contain grammar which is currently in practice not covered by categorical compositional method of Coecke et al. (2010).

Apart from handling a richer set of grammatical structures, the datasets contain a larger vocabulary, meaning that there are, for instance, more verbs for which matrices are needed to be build. Given an exhaustive experiments over a large number of parameter combinations it was not feasible to conclude such a study.

## 2.6 Conclusion

Similarity is an important notion in psychological theories of knowledge and behaviour. It is also useful in explaining language. Many NLP systems benefit from internal similarity components. However the exact definition of similarity is task-dependent: an IR system needs to know whether the words are related (for example, the verb *swim* is related to the noun *lake*), language models benefit from semantic similarity (the noun *lake* is similar to the noun *reservoir*, but not to the verb *swim*), dialog systems need to know the similarity of utterances by the role they play in discourse: *Hi!* and *Good morning.* are both greetings, for example.

Similarity measurement of words and multi-word units is theoretically challenging. Goodman (1972) argues that the similarity relation does not exist, because everything can be shown to be similar to everything else. Medin et al. (1993) and Markman and Gentner (1996) response that similarity has to be contextualised to be measured, that is, the features of interest need to be defined before the measurement.

According to Harris (1954), the word meaning does not need to be obtained to measure similarity, instead, the differences of occurrences of two words quantify the difference in their meaning. In this way, the problems with representing meaning in isolation raised by Frege are avoided, because word meaning is not constructed.

The principle of compositionality, that the representations of compounds are built from their parts, is the hallmark of categorical compositional semantics (Coecke et al. 2010). It extends the composition mechanism from formal semantics by replacing the symbolic representation of atoms and relations with tensors of various orders.

In categorical grammars, the meaning of a phrase is obtained by applying backward and forward application rules. Consider a phrase *John walks*. In this example, the string *John* is the atom *John′* of category $\mathsf{np}$ and type $e$, *walks* is the relation $\lambda x.\mathit{walks}'(x)$ of category $\mathsf{np}\backslash\mathsf{s}$ and type $(e \rightarrow t)$. During the phrase composition, the backward application rule is applied, so the category of the whole string becomes $\mathsf{s}$, its type is $t$ which is either *true* or *false* and the meaning is computed by the evaluation of the formula *walks*$'$(*John*$'$), basically the atom *John*$'$ is applied to the formula $\lambda x.$*walks*$'(x)$.

Categorical compositional semantics replaces atomic symbols in formal semantics with vectors (first-order tensors), and relations with higher-order tensors (so *walks* is represented by a second-order tensor, which is a matrix). To obtain the representation of a compound, tensor contraction is used instead of function application.

Word similarity (in the broad psychological sense) has been studied extensively: many datasets were proposed that focus on relatedness and similarity (in the specific sense that *lake* is similar to *reservoir*, but is not similar to *swim*). Several studies were carried out to identify the best parameter choice for the models of similarity.

Iterative parameter tuning is widely used to find the best model parameters. However, it does not take into account the interaction between the parameters of a similarity model. While some influential interactions are well known (for example, the weighting scheme interacts with the similarity score) and specific parameter choice is recommended (Positive PMI with cosine similarity), studies that introduce new parameters should not rely on existing recommendations, because an effective parameter combination might not be tested.

Phrase similarity has been studied much less than word similarity. Several datasets are proposed that focus on different aspects of similarity (for example, its application in word sense disambiguation within a context). Several compositional methods are developed. Other studied directions are the interaction of the lexical representations and compositional method and approximation of predicates to reduce the size of their representations.

However, the compositional methods were not evaluated in a systematic way to identify the best parameters (for example, the co-occurrence weighting function among the others) and rely on iterative tuning by taking as a starting point parameters that work best with words. This might lead to biased evaluation.

This thesis addresses the gap between the evaluation methodology of lexical and compositional similarity models. It is a systematic study of both words and phrasal similarity models. It covers a broad selection of parameter combinations that have not been tested before, especially on the phrase similarity tasks. It uses several datasets and employs a model selection methodology that is robust to overfitting.

# Chapter 3

# A methodology for robust parameter selection

Experiments with distributional vector space models can be divided into two classes. One class aims to achieve the highest score on a single task, or even a single dataset. Another class studies the behaviour of certain model parameters. The difference between the two classes can be expressed in the following questions:

- What parameter combination gives the highest result? Baroni et al. (2014b) is a representative study of this kind.
- Does a newly proposed technique outperform existing methods? For example, the study of Lapesa and Evert (2013), which contrasts ranked-based semantic priming estimation with distance-based.

The first question is applicable in a situation when conceptually different methods are compared, for example, the "count" and "predict" methods in Baroni et al. (2014b) or when the best performance score is required. The second question is applicable to a study of the difference in performance of parameters within a conceptual method, for instance, the comparison of neighbour rank and distance measure in predicting semantic priming of Lapesa and Evert (2013), where the goal is not to identify the best model, but to contrast parameter instances.

The co-occurrence information can be used in different ways to build distributional models of meaning (Turney and Pantel 2010). This has led to a series of systematic parameter studies (Baroni et al. 2014b, Bullinaria and Levy 2007; 2012, Kiela and Clark 2014, Lapesa and Evert 2014, Levy et al. 2015). All of them explore numerous parameter combinations to report the best scores and derive recommendations for the optimal parameter choice.

Lapesa and Evert (2014) make one step further in studying parameter behaviour by identifying the most influential parameters and their two-way interactions with a linear model, which is fitted so that parameters of a vector space model predict the performance of the vector space model on a task. It avoids iterative parameter tuning by testing all possible parameter combinations, so that unknown parameter interactions are captured. They avoid overfitting[1] and noise in the data by using a linear regression model.

The first goal of this work is to provide the representative performance measures of count-based distributional models of meaning—so that they could be compared to other semantic models. The second and the main goal is to study the general behaviour of vector space parameters and compositional operators—so that the compositional operators could be fairly compared. The study is performed systematically using recently developed evaluation datasets for lexical and phrasal similarity. We express the goal in two research questions that are related to the categories of studies stated above.

- What is the performance limit of distributional models of meaning?
- How do compositional operators and lexical representations affect one another?

## 3.1 Strategies for avoiding overfitting

This work adopts the strategy of Lapesa and Evert (2014) to avoid overfitting and reduce noise in parameter selection. Following them, we use several evaluation datasets one by one. In this case, we are able to identify situations when a model behaves particularly well on one dataset, but poorly on another, which is an example of overfitting.

In our case, overfitting (Dietterich 1995) might happen because a large number of models is being compared. The models are instantiated on a large, but limited in size linguistic resource and are evaluated on a limited number of datasets which are also limited in size. The goal is to capture a general phenomena of similarity of words and phrases, where the resources serve as proxies. Because a corpus is a sample of a language and the similarity datasets are samples of similarities, they might introduce biases that do not exists in language. Some words might be more similar according to the corpus, because it overrepresents a particular topic, also some words might be more similar due to imperfect similarity dataset construction protocol. Finally, random patterns might be introduced that taken into account might lead to higher performance.

While the source corpus and evaluation datasets are fixed, to minimise a chance of picking an overfitted model is to adopt evaluation procedure. The first action is to use several evaluation datasets and test models not only on the same training dataset, but also on one that is distinct from it. If the performance of a model dramatically drops, it is a sign of overfitting.

To be able to identify models that perform well on several datasets, we also test the models on all datasets simultaneously by aggregating model performance scores. We do not aggregate

[1] “Overfitting occurs when classifiers make decisions based on accidental properties of the training set that will lead to errors on the test set (or any new data)” Manning and Schuetze (1999).

dataset entries by, for example, taking a union of all of the entries in them because the judgements are on different scales and the participants were given different instructions.

The models are tested on two word similarity datasets: SimLex-999 (Hill et al. 2015) and MEN (Bruni et al. 2014). These two datasets are chosen because they are larger than other previously used datasets. Batchkarov et al. (2016) argue that the score variance is strongly dependent on the size of the evaluation dataset: the larger the dataset the more reliable experiment results are. SimLex-999 consists of 999 word pairs and MEN consists of 3 000 word pairs, making them the largest lexical datasets available to us. Other similar datasets are much smaller in size: 353 word pairs (Finkelstein et al. 2002) and 65 word pairs (Rubenstein and Goodenough 1965) for example.

The three phrasal datasets that are employed in this study are KS14 (Kartsaklis and Sadrzadeh 2014), GS11 (Grefenstette and Sadrzadeh 2011a) and PhraseRel (Section 5). They consist of phrases with controlled syntax (all of them are subject-verb-object phrases) and cover two relationships between phrases: similarity and relevance.

We test the models on the lexical datasets simultaneously to see whether there are models that perform well on both lexical datasets and thus avoid individual dataset characteristics that are independent to the phenomena of interest (similarity, relatedness or relevance). Similarly, we test the models on the phrasal datasets.

The scores on lexical and phrasal datasets are combined to identify a model that is universally good in lexical and compositional tasks. The model selection procedure is performed in two ways. First, we take the compositional operator into account, so we are able to recommend models that perform well on lexical and phrasal datasets with addition, multiplication and Kronecker (see Section 2.4.2 for the description of compositional operators). Finally, we abstract over the compositional operator and seek a model that achieves competitive results in both lexical and phrasal tasks with all operators.

We test the models on several datasets, transfer selected models to the unseen datasets and perform model selection on their combinations to avoid overfitting and obtain reliable model performance measurements.

In addition to that, we also report the results of the models that performed best in our exhaustive evaluation of testing all possible parameter combinations. This allows us to see whether overfitting actually happens, as we expect that during transfer the models with the highest scores will degrade in performance to a greater extent than the models selected more conservatively.

In the sections to follow, we discuss three parameter selection methods that we have applied.

### 3.1.1 Best model

This parameter selection technique chooses the parameters that yield the best result. This method is widely adopted. However, as previously discussed, it might be prone to overfitting.

### 3.1.2 Cross-validation

Cross-validation (Ney et al. 1997) is a widely used model selection method where parameter selection is based on the average performance of the training splits over several evaluation runs. Cross-validation splits the datasets to $N$ parts. Then $N$ runs are performed where each part is used as a testing split and the rest is used as a training split such that the $n$th run will use the $n$th part as a testing split. Training splits are used to tune parameters. The average performance over the $N$ testing splits is reported. Note that different model parameters might be used across testing splits.

Even though cross-validation avoids overfitting, its performance results are not comparable with the best model selection because they are based on averages over the testing splits. Moreover, existing datasets are not made with such an evaluation in mind (Faruqui et al. 2016), and there is no common agreement on how the datasets should be split to the training and testing parts.

### 3.1.3 Heuristics

This parameter selection is based on the average performance of the models where some parameters are fixed.

We look for the average model performance for every dimensionality (for lexical experiments) or for every operator-dimensionality combination (for compositional experiments) and a parameter of interest. Knowing the average performances of the values of the parameter of interest, we choose the value with the highest upper bound of the 0.95 confidence interval.

Because parameters influence model performance differently, the parameters are processed in order of their ablation (Lapesa and Evert 2014). A parameter's ablation is proportional to the reduction of the adjusted $R^2$ scores between a linear model that treats all parameters as independent variables and a linear model that leaves out the parameter of interest from the independent variables.

This method not only avoids overfitting but also yields evaluation results that are comparable with the best-model reports.

## 3.2 Parameters

This section explains in detail the parameters that are explored in the experiments. The core of the parameters are the parameters that modify the co-occurrence frequencies. The other parameters define the dimensionality of the vector space, the similarity measure and compositional operator.

### 3.2.1 Co-occurrence quantification

#### PMI variants (discr)

Most co-occurrence weighting schemes[2] in distributional semantics are based on *point-wise mutual information* (PMI, Equation 3.1, Church and Hanks (1989; 1990), Levy and Goldberg (2014), Turney and Pantel (2010)).

$$\text{PMI}(x, y) = \log \frac{P(x, y)}{P(x)P(y)} \tag{3.1}$$

PMI in its raw form is problematic: non-observed co-occurrences lead to infinite PMI values, making it impossible to compute similarity. A common solution to this problem is to replace all infinities with zeros, and we use PMI hereafter to refer to a weighting with this fix.

An alternative solution is to increment the probability ratio by 1, which also makes the weighted values non-negative; we refer to weighting scheme as *compressed PMI* (CPMI):

$$\text{CPMI}(x, y) = \log \left(1 + \frac{P(x, y)}{P(x)P(y)}\right) \tag{3.2}$$

Another issue with PMI is its bias towards rare events. Consider a context $c_r$ with very low probability (it could be a rare context word, a tokenization error or a misspelled word). The probability $P(c_r)$ will be much lower than for other contexts, and at the same time the PMI values for that feature will be higher. It is the same for rare target words, which due to the power-law distribution of tokens is pervasive. We refer to this issue as PMI's *Achilles heel*.

#### Shifted PMI (neg)

Many approaches use only *positive* PMI values, as negative PMI values may not positively contribute to model performance (Turney and Pantel 2010). This can be generalised to an additional cutoff parameter $k$ (abbreviated as neg) following Levy et al. (2015), giving our third PMI variant: *shifted PMI* or SPMI for short:

$$\text{SPMI}_\text{k} = \max(0, \text{PMI}(x, y) - \log k) \tag{3.3}$$

[2] We abbreviate this parameter as `discr` because the weighting scheme discriminates the features.

We can apply the same idea to CPMI and obtain *shifted compressed PMI* or SCPMI:

$$\mathrm{SCPMI_k} = \max(0, \mathrm{CPMI}(x, y) - \log 2k) \tag{3.4}$$

### Frequency weighting (freq)

One way of solving PMI's bias toward rare events is to weight the value by the co-occurrence frequency obtaining the *local mutual information* (LMI, Evert (2005)), for clarity we refer to LMI as nPMI:

$$\mathrm{nPMI}(x, y) = n(x, y)\,\mathrm{PMI}(x, y) \tag{3.5}$$

where $n(x, y)$ is the number of times $x$ was seen together with $y$. We refer to $n$-weighted PMIs as nPMI, nSPMI, etc. When this weighting component is set to 1, it has no effect; we can explicitly label it as 1PMI, 1SPMI, etc. In addition to the extreme $1$ and $n$ weightings, we also experiment with the $\log n$ weighting. We refer to this parameter as `freq`.

### Context distribution smoothing (cds)

Levy et al. (2015) show that performance is affected by smoothing the context distribution $P(x)$:

$$P_\alpha(x) = \frac{n(x)^\alpha}{\sum_{f \in F} n(f)^\alpha} \tag{3.6}$$

where $n(x)$ is the frequency of the term $x$, F is the set of the features in the co-occurrence matrix and $n(f)$ is the frequency of the feature in the corpus. We experiment with $\alpha = 1$ (no smoothing) and $\alpha = 0.75$. We call this estimation method *local context probability*.

Recchia and Nulty (2017) investigate an optimal choice of $\alpha$. They notice when $\alpha = 1$ then $P_1(x)$ highly correlates with the frequency of the term $x$, while when $\alpha = 0$ then $P_0(x) = 1$ and correlates inversely with the frequency of the term $x$. Moreover, smoothed PMI is closely related to another measure $\mathrm{SCI}(x, y) = \frac{P(x,y)}{P(x)\sqrt{P(y)}}$ (Washtell and Markert 2009), which performs much more poorly (Recchia and Nulty 2017).

They precede with hypothesising that $\alpha = 0.75$ neither positively nor negatively correlates with the with the word frequency. Their experiments show that the value of $\alpha = 0.77$ that minimises the absolute value of the measure's correlation to the word frequency are not far off from the values of $\alpha = 0.765$ that maximises correlations to human judgements.

We also estimate a *global context probability* based on the size of the corpus $|C|$:

$$P(x) = \frac{n(x)}{|C|} \tag{3.7}$$

### Quantification measure generalisation

To systematically study the aforementioned quantification measures, together with other variations, we propose to view all these measures as instances of this general formula:

$$\text{Quantification}(x, y) = \text{freq}(x, y)\,\text{discr}(x, y) \tag{3.8}$$

which consists of two components: $\text{freq}(x, y)$ which quantifies the co-occurrence of two terms—a target term $x$ and a feature term $y$; and $\text{discr}(x, y)$ which quantifies the "surprise" or "informativeness" of seeing (or not seeing) the two terms together, labeled as discriminativeness.

In this framework, PMI can be seen as a quantification measure where the frequency component is the constant 1 and the discriminativeness is the PMI itself. SPMI, CPMI and SCPMI are seen analogously. For nPMI, $\text{freq}(x, y) = n(x, y)$ and $\text{discr}(x, y) = \text{PMI}(x, y)$.

From the probabilistic point of view, under the independence assumption of two words occurring together, nPMI can be interpreted as measuring the logarithm of the ratio of the probabilities of groups of length $n$ (the group that contains only pairs of $(x, y)$s and another one that contains $x$s and $y$s):

$$n \log \frac{P(x, y)}{P(x)P(y)} = \log \frac{P(x, y)^n}{P(x)^n P(y)^n} \tag{3.9}$$

From the geometric point of view, the transformation from 1PMI to nPMI changes the directions of vectors by pulling the vectors toward the dimensions for which $n(x, y)$ is higher. As a side effect, it also stretches the vectors.

From the linguistic perspective, 1PMI captures the tendency for a word to co-occur with another word in general (captured by the direction of a vector), while nPMI captures the expectation of seeing a particular co-occurrence in the source corpus. This is encoded in both the direction and the length of a vector.

## 3.2.2 Other model parameters

The source corpus that we use is the concatenation of ukWaC and Wackypedia (Ferraresi et al. 2008).[3] A window of 5 neighbour words from each side is used to collect co-occurrences.

### Vector dimensionality ( D )

As context words we select the 1K, 2K, 3K, 5K, 10K, 20K, 30K, 40K and 50K most frequent lemmatised nouns, verbs, adjectives and adverbs in the source corpus. All context words are part-of-speech tagged, but we do not distinguish between refined word types (e.g. intransitive vs. transitive versions of verbs).

[3] The ukWaC corpus is available at http://wacky.sslmit.unibo.it.

| Parameter | Abbreviation | Values |
|---|---|---|
| Dimensionality | $D$ | 1K, 2K, 3K, 5K, 10K, 20K, 30K, 40K, 50K |
| Discriminativeness | `discr` | PMI, CPMI, SPMI, **SCPMI** |
| Frequency weighting | `freq` | 1, $n$, $\boldsymbol{\log n}$ |
| Shifting | `neg` | **0.2**, **0.5**, **0.7**, 1, 1.4, 2, 5, 7 |
| Context distribution smoothing | `cds` | ***global***, 1, 0.75 |
| Similarity | | Cosine, Correlation and Inner product |
| Window size | | 5 from both sides |
| Corpus | | Concatenation of ukWaC and Wackypedia |

**Table 3.1:** Model parameters and their values used in the experiments. To our knowledge, values that are in bold have not been used previously.

### Similarity measure

To be able to measure the similarity of two words, we need to be able to compare their vectors.[4] A very high-level approach is to look at how two words agree on their features. If two-word vectors tend to have approximately equal values for most of their components, then this is a good indication of the similarity of the words they represent.

The cosine of the angle between two vectors is a widely used similarity measure in distributional semantics (Lapesa and Evert 2014, Turney and Pantel 2010).

$$\cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \|\vec{y}\|} \tag{3.10}$$

However, the inner product $\vec{x} \cdot \vec{y}$ is preferred in information retrieval and current state-of-the-art natural language processing systems (Levy et al. 2015, Mikolov et al. 2013b;c). The cosine of the angle is the inner product of the normalised vectors (using Euclidean $L_2$ length).

Normalisation reduces all vectors to unit length leaving their directions to characterise them. Thus, remembering that vector length depends on overall frequency, linguistically we have two measures: the cosine measure that is concerned with similarity, and inner product with no normalisation which in addition to similarity also reflects word frequency and expectation factors. If the vectors are normalised, then the inner product and the cosine measures are the same.

An advantage of cosine in a lexical similarity task is that it does not depend on the word frequency. Imagine a situation where the similarity of a frequent and a rare word is calculated, it will be lower than the similarity between two frequent words. Then the similarity judgment should not depend on the relative frequency of the words; instead, their tendency of agreement on features should take the dominant role.

For example, nPMI makes "feature selections" by weighting PMI values with the co-occurrence frequency, as discussed in Section 3.2.1. When cosine is applied, the stretching effect of nPMI

[4] Even though similarity is not strictly a parameter of a distributional model, it is treated the same as a model parameter.

| Method | Linear algebraic formula | Reference |
|---|---|---|
| Addition | $\overrightarrow{Sbj} + \overrightarrow{Verb} + \overrightarrow{Obj}$ | Mitchell and Lapata (2008) |
| Multiplication | $\overrightarrow{Sbj} \odot \overrightarrow{Verb} \odot \overrightarrow{Obj}$ | Mitchell and Lapata (2008) |
| Kronecker | $\widetilde{Verb} \odot (\overrightarrow{Sbj} \otimes \overrightarrow{Obj})$ | Grefenstette and Sadrzadeh (2011b) |
| Relational | $\overline{Verb} \odot (\overrightarrow{Sbj} \otimes \overrightarrow{Obj})$ | Grefenstette and Sadrzadeh (2011a) |
| Copy-object | $\overrightarrow{Sbj} \odot (\overline{Verb} \times \overrightarrow{Obj})$ | Kartsaklis et al. (2012) |
| Copy-subject | $\overrightarrow{Obj} \odot (\overline{Verb}^{\mathsf{T}} \times \overrightarrow{Sbj})$ | Kartsaklis et al. (2012) |
| Frob. add. | $\left(\overrightarrow{Sbj} \odot (\overline{Verb} \times \overrightarrow{Obj})\right) + \left(\overrightarrow{Obj} \odot (\overline{Verb}^{\mathsf{T}} \times \overrightarrow{\mathrm{Sbj}})\right)$ | Kartsaklis and Sadrzadeh (2014) |
| Frob. mult. | $\left(\overrightarrow{Sbj} \odot (\overline{Verb} \times \overrightarrow{Obj})\right) \odot \left(\overrightarrow{Obj} \odot (\overline{Verb}^{\mathsf{T}} \times \overrightarrow{Sbj})\right)$ | Kartsaklis and Sadrzadeh (2014) |
| Frob. outer | $\left(\overrightarrow{Sbj} \odot (\overline{Verb} \times \overrightarrow{Obj})\right) \otimes \left(\overrightarrow{Obj} \odot (\overline{Verb}^{\mathsf{T}} \times \overrightarrow{Sbj})\right)$ | Kartsaklis and Sadrzadeh (2014) |

**Table 3.2:** Compositional operators

is eliminated, but the rotational effect stays. On average, the rotational effect will be much more significant for rare words, while frequent words are more likely to be stretched.

In addition to cosine and inner product, we use correlation (Kiela and Clark 2014) to measure similarity:

$$\text{correlation}(\vec{x}, \vec{y}) = \frac{(\vec{x} - \bar{x}) \cdot (\vec{y} - \bar{y})}{\|(\vec{x} - \bar{x})\| \|(\vec{y} - \bar{y}\|} \tag{3.11}$$

where $\bar{x}$ is the mean of the elements of $\vec{x}$ and $\bar{y}$ is the mean of the elements of $\vec{y}$.

### Compositional operator

For phrasal tasks, the phrase vectors are obtained via composition of the phrase constituents' vectors using addition, multiplication (Mitchell and Lapata 2008; 2010), Kronecker (Grefenstette and Sadrzadeh 2011a) and tensor-based operators (Coecke et al. 2010, Grefenstette and Sadrzadeh 2011a, Kartsaklis and Sadrzadeh 2014, Kartsaklis et al. 2012). Table 3.2 lists the operators used in this study.

As a non-compositional baseline, we take the dummy operator `head`, which ignores the subject and the object of a phrase, causing the vector of a whole phrase to be equal to the vector of its verb.

## 3.3 Hypotheses

To conduct the study, we introduce hypotheses that reflect the current state in the field of distributional semantics and facilitate the answering of the research questions.

### 3.3.1 General

**Hypothesis 1 (H1):** *Heuristics-based model selection avoids overfitting.*

We expect that models that are chosen using heuristics achieve better results on the datasets they were not instantiated on than the best models.

**Hypothesis 2 (H2):** *The relative difference between the score of the best model and the score of the model selected using heuristics is less than 10%.*

The optimal results reported in Lapesa and Evert (2014, Table 5) are within the 10% margin (with an exception of the ESSLLI dataset, where the margin is 21%). We expect similar relative differences in our results.

**Hypothesis 3 (H3):** *High-dimensional models are more likely to perform better than their low-dimensional counterparts.*

As the vector space dimensionality increases, performance stabilises (Bullinaria and Levy 2012, Kiela and Clark 2014, Lapesa and Evert 2014). We speculate that in a high-dimensional case the difference between parameter choices matters less and thus higher results are reported more often for high-dimensional spaces.

**Hypothesis 4 (H4):** *There is a universal model that performs well on a broad range of tasks.*

We are interested to see whether there is one parameter choice that performs competitively on all tasks. The results of Lapesa and Evert (2014) show that there is a general model whose performance is close to dataset- and task-specific models. We expect this to be the case also for compositional models.

### 3.3.2 Parameter dependence on dimensionality

**Hypothesis 5 (H5):** *The optimal parameter choice depends on dimensionality.*

We expect that the co-occurrence counts of the most frequent pairs do not contain noise (note that dimensions are ranked by the frequency of the corresponding features, so by design the low-dimensional vector spaces incorporate the most frequent features). The counts, and therefore the probability estimates, of less frequent pairs are noisy and require a special treatment to compensate for PMI's Achilles heel when small co-occurrence counts lead to extremely high PMI values (Levy et al. 2015).

**Hypothesis 6 (H6):** $N$ *and* $\log n$ *frequency components are beneficial for high-dimensional spaces.*

This is the most direct way of boosting high co-occurrence counts (Evert 2005). nPMI is shown to be a good choice in lexical tasks (Bruni et al. 2012).

**Hypothesis 7 (H7):** *Low-dimensional spaces do not need context distribution smoothing, while high-dimensional spaces benefit from it.*

This is because the estimated probabilities of rare contexts are noisy. This smoothing is shown successfully in high-dimensional count-based models (Levy et al. 2015) and word2vec (Mikolov et al. 2013a). However, these recommendations were not tested on low-dimensional vector spaces, that are widely used by compositional models.

**Hypothesis 8 (H8):** *Low-dimensional spaces benefit from being dense, while high-dimensional spaces benefit from being sparse.*

Sparsity is controlled by the shifting parameter $k$; lower $k$ values make vectors denser.

**Hypothesis 9 (H9):** *Cosine is an optimal similarity measure for low-dimensional spaces, while correlation is for high-dimensional spaces.*

Correlation is shown to be the best choice by Kiela and Clark (2014) while cosine is generally perceived as the best similarity metric. It might be the case that the standardisation of vector values by subtracting the mean is effective for high-dimensional spaces.

### 3.3.3 Lexical

**Hypothesis 10 (H10):** *In lexical tasks, there should be little difference between PMI and its compressed version CPMI.*

The main effect of CPMI is to transform negative values into the positive range of $(0; 1)$. One of the reasons to avoid negative values is that they might be problematic for multiplication during composition, as the sign of the result depends on the number of negative components. However, as there is no composition involved in lexical tasks, the weighting schemes should behave equally.

### 3.3.4 Compositional

**Hypothesis 11 (H11):** *Models that perform well on lexical tasks also perform well on compositional tasks.*

If this is the case, then iterative tuning of parameters is justified. As an important consequence, the studies that perform evaluation of compositional models do not need to explore all possible parameter combinations.

**Hypothesis 12 (H12):** *The best models for compositional tasks should take word order into account.*

We expect that the word order sensitive models outperform the models that ignore word order.

**Hypothesis 13 (H13):** *Compositional methods that include addition perform best with either PMI or SPMI, while methods that include multiplication should work best with CPMI or SCPMI.*

One of the reasons for this is the presence of negative values. For example, in the case of multiplication as a compositional operator, the sign of a vector component depends on the number of the corresponding negative components of the constituents. If the number of negative values is odd, then the resulting value will be negative—this makes the difference between 0.001 and -0.001 significant. In the first case, the value means that the co-occurrence pair is weakly associated. But in the second case, the value means that the co-occurrence is weakly unassociated. This also applies to categorical operators where the signs of the result vector components depend on the number of multiplication operations that leads to them.

# Chapter 4

# Relationships between words

This chapter describes experiments to discover optimal parameters for lexical similarity, and the behavioural patterns of the parameters.[1] Simlex-999 (Hill et al. 2015) and MEN (Bruni et al. 2014) are the two datasets that are used for evaluation. They provide averaged values of similarity judgements between pairs of words.

The experiment results and selected parameters are reported on the datasets individually. Then, the model selection is performed on a combination of the two datasets. The results are selected using three methods: Max selection, where the best score is selected, cross-validation, which separates model selection and score computation, and a selection based on heuristics, here parameters are chosen by their influence.

The experiments show that while the best parameter selection depends on the dataset, there is a global optimal parameter selection that is good on all datasets. In general, we find that non-constant frequency component, context distribution smoothing and shifting should be used for high-dimensional spaces to compensate for the noise that is introduced by a large number of context features.

## 4.1 Experiments on SimLex-999 dataset

SimLex-999 is a dataset to evaluate lexical semantic models (Hill et al. 2015). It tests how well a model captures similarity between word pairs. The dataset implicitly distinguishes similarity and relatedness, so related pairs such that *coffee*, *cup* are not considered to be similar in this dataset. It consists of 666 noun-noun pairs, 222 verb-verb pairs and 111 adjective-adjective pairs. The models are evaluated by computing the Spearman's-$\rho$ (the correlation of ranked model predictions with ranked human judgements). Minimum required difference for signif-

[1] Much of the work in this chapter has appeared as Milajevs et al. (2016) at the ACL Student Workshop 2016.

| dimensionality | SimLex999 | freq | discr | cds | neg | similarity |
|---:|---:|---|---|---|---:|---|
| 1 000 | **0.369** | 1 | spmi | 1 | 0.2 | inner_product |
| 2 000 | **0.389** | 1 | scpmi | global | 0.7 | inner_product |
| 3 000 | **0.376** | 1 | spmi | 0.75 | 0.2 | inner_product |
| 5 000 | 0.363 | logn | scpmi | global | 1.0 | cos |
| 10 000 | **0.371** | logn | scpmi | 1 | 0.7 | cos |
| 20 000 | **0.381** | logn | scpmi | 0.75 | 0.7 | cos |
| 30 000 | **0.383** | logn | scpmi | 0.75 | 0.7 | cos |
| 40 000 | **0.384** | logn | scpmi | 0.75 | 0.7 | cos |
| 50 000 | **0.385** | logn | scpmi | 0.75 | 0.7 | cos |

**Table 4.1:** SimLex-999 Max selection. Correlation values that are not statistically significant from the highest value of 0.389 are in bold. For SimLex-999, $\sigma^{0.9}_{0.05} = 0.023$ making the values greater than 0.366 indistinguishable from the highest result.

icance (MRDS, Rastogi et al. (2015)) is used for statistical significance testing. For SimLex-999, $\sigma^{0.9}_{0.05} = 0.023$.[2]

### 4.1.1 Max selection

Figure 4.1 illustrates the results based on the best model selection and Table 4.1 shows the results together with chosen parameters. Note that maximum selection is identical with cross-validation: they pick the same models.

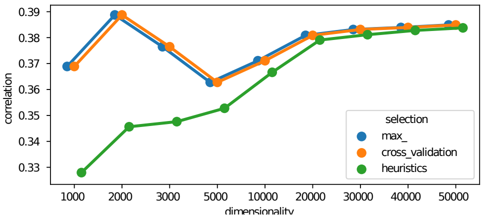


**Figure 4.1:** SimLex-999 results

In general, model performance increases as dimensionality increases. There is no statistically significant difference between the scores with an exception of the 5 000 model, which underperforms. The best result of 0.389 is achieved with a 2 000 dimensional space. Model performance becomes stable for dimensions greater than 20 000, which suggests that rare context features contribute very little to similarity estimation.

As we see later in Section 4.3.1 the best result is an example of overfitting. The 2 000 dimensional model performs very well on SimLex-999 achieving the score of 0.389, but underperforms on MEN achieving the score of 0.660, while the model chosen using heuristics yields the score of 0.724 on MEN (the difference is statistically significant). In addition, the model selected by using heuristics is not statistically significantly different from the maximum result for this dimensionality (0.728, Figure 4.8a).

For spaces with dimensionality less than 5 000, the frequency parameter set to 1 and the inner product as the similarity measure yield the best results. Otherwise, cosine with lognSCPMI, smoothing $\alpha = 0.75$ and shifting $k = 0.7$ gives the best results. This supports our hypothe-

[2] The results are available at http://www.eecs.qmul.ac.uk/~dm303/thesis/results_all.csv.

| dimensionality | SimLex999 | freq | discr | cds | neg | similarity |
|---:|---:|---|---|---|---|---|
| 1 000 | 0.328 | logn | scpmi | 1 | 0.7 | correlation |
| 2 000 | 0.346 | logn | scpmi | 1 | 0.7 | correlation |
| 3 000 | 0.348 | logn | scpmi | 1 | 0.7 | correlation |
| 5 000 | 0.353 | logn | scpmi | 1 | 0.7 | correlation |
| 10 000 | **0.367** | logn | scpmi | 0.75 | 0.7 | correlation |
| 20 000 | **0.379** | logn | scpmi | 0.75 | 0.7 | correlation |
| 30 000 | **0.381** | logn | scpmi | 0.75 | 0.7 | correlation |
| 40 000 | **0.383** | logn | scpmi | 0.75 | 0.7 | correlation |
| 50 000 | **0.384** | logn | scpmi | 0.75 | 0.7 | correlation |

**Table 4.2:** SimLex-999 selection based on heuristics. The highest value is 0.384. The values that are greater than 0.361 are indistinguishable from the highest score.

ses that high-dimensional spaces benefit from a non-constant frequency (H6), smoothing of context distribution (H7) and the sparsity of vectors (H8).

### 4.1.2 Heuristics

Analysis of variance is used to obtain parameter influence. First, we fit a linear regression model that takes into account all model parameters. The similarity score is a dependent variable, while the parameters of a similarity model are independent variables. In addition to individual parameter influence, we also consider their two-way interactions. The full linear model is defined as:

$$\mathit{score} \sim \sum_{p \in P} p + \sum_{p,p' \in P \times P} p \cdot p' \tag{4.1}$$

where $P$ is the set of parameters, in our case $P = \{\mathit{freq}, \mathit{discr}, \mathit{cds}, \mathit{neg}, \mathit{similarity}\}$.

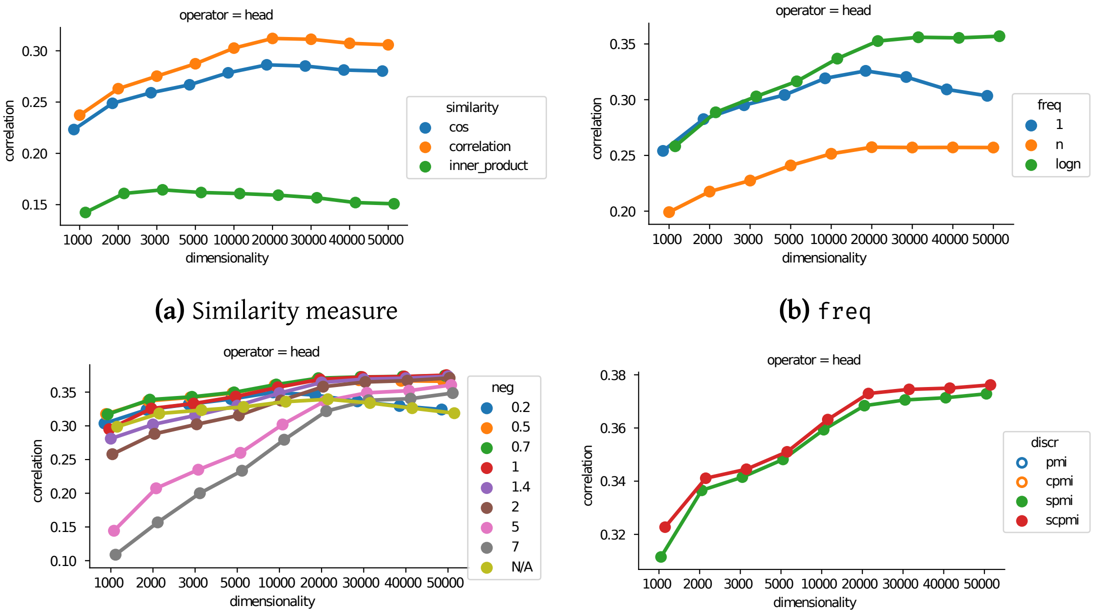
The linear model achieves an adjusted $R^2$ value of 0.867, indicating that the linear model is able to predict the performance of a similarity model based on the parameters $P$ quite well.

After fitting a model on the full set of parameters, we fit six “restricted” linear models, each of them excluding a parameter $\bar{p}$:

$$\mathit{score} \sim \sum_{p \in P \setminus \{\bar{p}\}} p + \sum_{p,p' \in (P \setminus \{\bar{p}\}) \times (P \setminus \{\bar{p}\})} p \cdot p' \tag{4.2}$$

We compute $R^2$ scores of the models that exclude a parameter. The difference of the $R^2$ sore of the full model and the $R^2$ score of a restricted model is the partial $R^2$ score of the excluded parameter. Table 4.3 shows partial $R^2$ scores for all the parameters. The most influential parameters, in decreasing order, are similarity, `freq` and `neg`, the contribution of other parameters to model performance estimation is minimal.

To identify whether one parameter choice outperforms another, the mean performance of both is estimated and the difference between them is compared. If the difference is less than



**(a)** Similarity measure

**(b)** `freq`

**(c)** `neg`

**(d)** `discr`

**Figure 4.2:** SimLex-999 influence of the similarity measure, `freq`, `neg` and `discr`. For the values of minimal required difference for significance (MRDS) refer to Table 2.3.

the MRDS for the dataset, then the parameter choices are statistically indistinguishable.

Figure 4.2a shows the average performance of similarity measures. Correlation outperforms all other measures for all dimensions and peaks at the dimensionality of 20 000, as Table 4.2 shows. However, for $D < 10\,000$, the difference between correlation-based and cosine-based similarity measures is not statistically significant, while it is for $D \geqslant 10\,000$. This supports H9 that correlation performs well with high-dimensional spaces.

| parameter | partial $R^2$ |
|---|---|
| similarity | 0.379 |
| freq | 0.268 |
| neg | 0.241 |
| dimensionality | 0.084 |
| discr | 0.077 |
| cds | 0.064 |

**Table 4.3:** SimLex-999 feature ablation

The influence of `freq`, the second parameter, is shown in Figure 4.2b. $\log n$ frequency outperforms other choices for all dimensions. At 20 000 and more dimensions, $\log n$ statistically significantly outperforms 1. Also, $\log n$'s performance stabilises: variance decreases and the performance stays constant. Taking into account that for $D < 20\,000$ the difference between 1 and $\log n$ is not statistically significant, H6 is supported in this case as well, suggesting that for low-dimensional spaces the frequency component is unnecessary.

The third parameter neg, with a value of 0.7, shows the best performance (Figure 4.2c), but 0.5 and 1 are statistically indistinguishable from it. With more than 3 000 dimensions, the difference between 0.7 and 1.4 is statistically insignificant. Similarly, 2 is not statistically different with more than 10 000 dimensions and 5 with more than 30 000 dimensions.

The performance of neg set to 7 increases as dimensionality increases, but the difference with 0.7 is statistically significant for all dimensions. We expect that with more dimensions, the difference between 0.7 and 7 becomes statistically insignificant.

Models with neg set to 0.2 become statistically different with more than 30 000 dimensions.

The score variance is much lower for high-dimensional spaces than for low-dimensional spaces, so for high-dimensional more choice of neg are statistically indistinguishable from the best choice of 0.7 This gives support to H3 that high-dimensional models are more likely to outperform low-dimensional models.

Also, lower $k$ values, which make vectors denser, lead to higher performance in low-dimensional spaces, supporting H8 that expects low-dimensional models benefit from dense vectors.

Models that do not perform shifting (abbreviated as N/A in Figure 4.2c), peak at 20 000 dimensions and decrease afterwards with increasing variance. With 10 000 dimensions or more they are statistically different from the best result.

There is no statistically significant difference between SPMI and SCPMI performance with a slight advantage to SCPMI (Figure 4.2d): PMI value compression into the range of $(0; 1)$ is unnecessary for lexical tasks (H10).

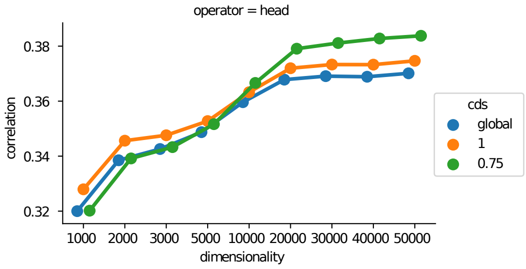


**Figure 4.3:** SimLex-999 influence of cds

Finally, models benefit from context distribution smoothing; spaces with less than 10 000 dimensions produce the best results with $\alpha = 1$. For spaces with higher dimensionality, $\alpha = 0.75$ is the most advantageous (Figure 4.3). However, the difference between $\alpha = 1$ and $\alpha = 0.75$ is statistically insignificant. H7 (smoothing is beneficial for high-dimensional spaces) is not supported.

### 4.1.3 Difference between Max selection and heuristics on SimLex-999

As expected, manual, heuristic-based parameter selection is more homogeneous, as Table 4.2 shows. Both selection models agree on parameters for high-dimensional spaces ($D \geqslant 2\,000$), with an exception of similarity: Max selection prefers cosine, while manual prefers correlation-based similarity measures. Because of this, manual selection does not pick the

| dimensionality | men | freq | discr | cds | neg | similarity |
|---:|---:|---|---|---|---:|---|
| 1 000 | 0.686 | 1 | scpmi | global | 1.4 | correlation |
| 2 000 | 0.728 | logn | scpmi | 1 | 0.7 | cos |
| 3 000 | 0.737 | logn | scpmi | 1 | 0.7 | cos |
| 5 000 | 0.743 | logn | scpmi | 0.75 | 0.7 | cos |
| 10 000 | **0.753** | logn | scpmi | 0.75 | 1.0 | correlation |
| 20 000 | **0.763** | logn | scpmi | 0.75 | 1.0 | correlation |
| 30 000 | **0.765** | logn | scpmi | 0.75 | 1.0 | correlation |
| 40 000 | **0.765** | logn | scpmi | 0.75 | 1.0 | correlation |
| 50 000 | **0.765** | logn | scpmi | 0.75 | 1.0 | correlation |

**Table 4.4:** MEN Max selection. Correlation values that are not statistically significant from the highest value of 0.765 are in bold. For MEN, $\sigma^{0.9}_{0.05} = 0.013$ making the values greater than 0.752 indistinguishable from the highest result.

best result for the 2 000 dimensional model, but at 50 000 dimensions a model selected manually scores only 0.001 lower: 0.384 versus 0.385 as also seen on Figure 4.1.

The average relative difference between Max selection and heuristics is 0.039 (3.9%), which is within the 10% margin set by H2. Moreover, statistically significant difference between the results is only for the models with dimensionality of 1 000, 2 000 and 3 000.

## 4.2 Experiments on MEN dataset

The MEN test collection consists of 3 000 word pairs judged for similarity (Bruni et al. 2014). In contrast to SimLex-999, the dataset does not distinguish between similarity and relatedness. As with SimLex-999, the models are evaluated by the Spearman's-$\rho$ correlation. The minimum significant difference for MEN $\sigma^{0.9}_{0.05} = 0.013$.

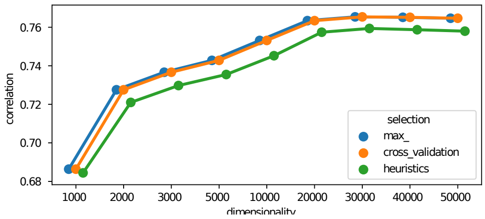


**Figure 4.4:** MEN results

### 4.2.1 Max selection

Figure 4.4 shows the selection results. Again, cross-validation results are identical with Max selection. Table 4.4 shows the results together with the selected models.

Model performance monotonically increases as dimensionality increases. The highest score of 0.765 is achieved by 3 spaces with $D \geqslant 30\,000$, lognSCPMI, smoothed context distribution ($\alpha = 0.75$), shifted PMI values ($k = 1$) and the similarity measure based on correlation. Models with the same parameter choice, but lower dimensionality (10 000 and 20 000) are statistically indistinguishable from the highest scoring models.

| dimensionality | men | freq | discr | cds | neg | similarity |
|---:|---:|---|---|---|---|---|
| 1 000 | 0.684 | logn | spmi | global | 2 | correlation |
| 2 000 | 0.721 | logn | spmi | global | 2 | correlation |
| 3 000 | 0.730 | logn | spmi | global | 2 | correlation |
| 5 000 | 0.735 | logn | spmi | global | 2 | correlation |
| 10 000 | 0.745 | logn | spmi | global | 2 | correlation |
| 20 000 | **0.757** | logn | spmi | global | 5 | correlation |
| 30 000 | **0.759** | logn | spmi | global | 5 | correlation |
| 40 000 | **0.759** | logn | spmi | global | 5 | correlation |
| 50 000 | **0.758** | logn | spmi | global | 5 | correlation |

**Table 4.5:** MEN selection based on heuristics. The highest value is 0.759. The values that are greater than 0.746 are indistinguishable from the highest score.

In comparison with SimLex-999, models with "more extreme" parameters give better results on MEN. For example, $\alpha = 0.75$ is the best for models tested on SimLex-999 with dimensionality starting with 20 000, while for models tested on MEN, this parameter choice is the best starting with 5 000. Similar behaviour is observed for neg and similarity. For high-dimensional spaces, the switch from SimLex-999 to MEN changes the best neg choice from 0.7 to 1 and similarity from cosine to correlation. Such a difference in parameter choices might suggest the difference between *relatedness* and *similarity*, but it still supports H6, H7 and H8 that frequency, context distribution smoothing and sparsity are important for high-dimensional spaces. H9 is also supported, as correlation is the best similarity measure for high-dimensional spaces.

## 4.2.2 Heuristics

The linear model gives an adjusted $R^2$ of 0.733, which is lower than on SimLex-999, but is still high. Table 4.6 shows partial $R^2$ scores for the explored parameters. The most influential parameter is neg, followed by freq and similarity. This is different from the case of SimLex-999, where the parameter's influence "order" is reversed.

| parameter | partial $R^2$ |
|---|---:|
| neg | 0.309 |
| freq | 0.204 |
| similarity | 0.183 |
| discr | 0.119 |
| dimensionality | 0.108 |
| cds | 0.086 |

**Table 4.6:** MEN feature ablation

The parameter neg, which controls sparsity, with $k = 2$ is preferable for spaces with dimensionality less than 20 000. For spaces with more dimensions, $k = 5$ is more beneficial (Figure 4.5a). Models without shifting and with $k$ set to 0.2, 0.5, 0.7 and 7 are statistically significantly different from the best choice for all dimensions. This replicates the suggestions of Levy et al. (2015). We, however, expect that for spaces with more than 50 000 dimensions even higher values should be preferred.

The choice of $k = 1$ with $D \geqslant 3\,000$ is statistically indistinguishable from the best choice, $k = 1.4$ is statistically indistinguishable with $D \geqslant 20\,000$, $k = 2$ is statistically indistinguishable for $2\,000 \leqslant D \leqslant 40\,000$ and $k = 5$ is statistically indistinguishable when $D \leqslant 10\,000$.

The choice of $k$ for MEN contrasts with the heuristics derived from SimLex-999, where the neg values of 0.5, 0.7 and 1 are statistically indistinguishable from each other, but still complies with H8: sparse high-dimensional spaces outperform their high-dimensional, but dense counterparts. The performance of the models with less than 20 000 dimensions is statistically different.

Regarding the frequency component, $\log n$ outperforms all other choices (Figure 4.5b). With 2 000 or more dimensions it is statistically significantly different from the constant frequency. It is always statistically significantly different from the linear frequency. H6—that frequency is needed for high-dimensional spaces—is once again confirmed.

Correlation is the preferred similarity measure (Figure 4.6a) it is statistically significantly different from both cosine and the inner product. This is, again, in line with the choice based on SimLex-999. However, the gap between cosine and correlation similarities stays constant, with an exception of $D = 1\,000$, where the gap is smaller, giving a weak support to H9: correlation performs best with high-dimensional spaces.

Overall, SPMI is the preferred discriminativeness (Figure 4.6b), however, it is closely followed by CPMI (the statistically significance difference for $D \leqslant 2\,000$) and SCPMI (is statistically indistinguishable for $3\,000 \leqslant D \leqslant 10\,000$). This contrasts with SimLex-999, where SCPMI is preferred. However, in both cases, the difference between the two choices is minimal. This is consistent with H10 that lexical models do not need PMI compression.

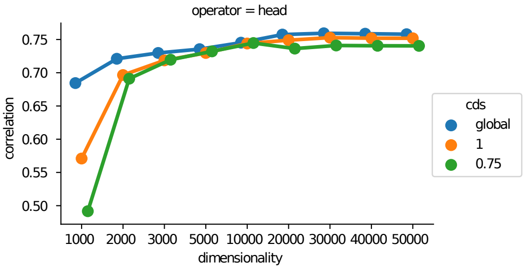
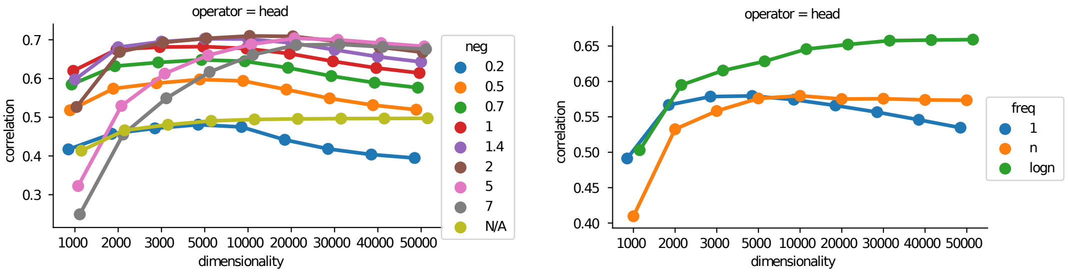


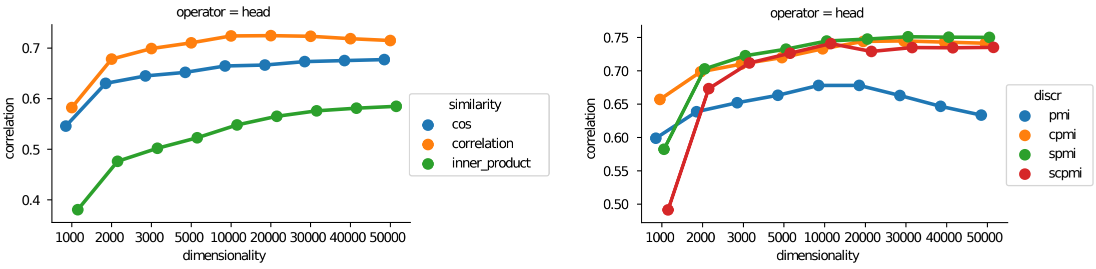
**Figure 4.7:** MEN influence of cds

Global context probability gives on average higher results for MEN (Figure 4.7). For models with 3 000 or more dimensions, local probabilities are statistically indistinguishable. For models with $2\,000 < D < 20,000$, $\alpha = 0.75$ is statistically indistinguishable as well.



**(a)** neg **(b)** freq

**Figure 4.5:** MEN influence of neg and freq



**(a)** similarity

**(b)** `discr`

**Figure 4.6:** MEN influence of similarity and `discr`

Note that SimLex-999 prefers context distribution smoothing (Figure 4.3). The difference in performance between local context probabilities and global context probabilities decreases as dimensionality increases, making a weak support of H7 that high-dimensional spaces benefit from context distribution smoothing.

### 4.2.3 Difference between Max selection and heuristics on MEN

The two selection procedures agree on fewer parameters than the ones based on SimLex-999. Both agree on discrimination (log $n$) and similarity score for spaces with dimensionality greater than 10 000 (correlation).

While SCPMI is chosen by Max selection, SPMI is preferred by the selection based on heuristics, however, the difference between the two is minimal, especially for $3\,000 \leqslant D \leqslant 10\,000$, where the difference is statistically in significant.

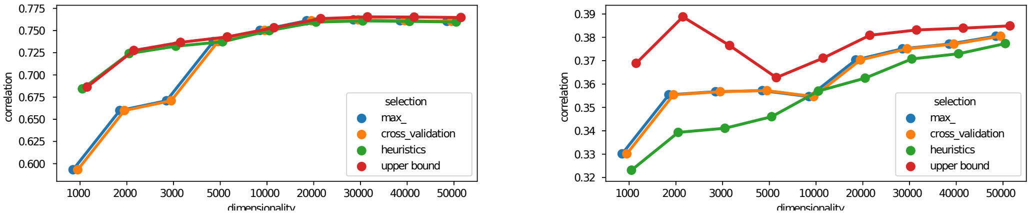
In contrast to the Max selection, which chooses the models with context distribution smoothing, heuristics prefers models with global context probabilities. However, with $3\,000 < D < 20\,000$, there is no statistically significant difference between `cds` values.

Also, heuristics picks models with higher shifting values $k$ (2 and 5), in contrast to Max selection, where 0.7 and 1 are chosen. Table 4.5 summarises the parameter selection based on heuristics.

The average relative difference between Max selection and heuristics is 0.008 (0.8%), supporting H2: the difference between Max selection and heuristics is within the 10% margin. Moreover, the difference in scores between the two selections is statistically insignificant.

### 4.2.4 Difference between heuristics based on MEN and SimLex-999

Heuristics based on MEN agree with ones based on SimLex-999 for two parameters: frequency (log $n$) and similarity (correlation). The methods disagree on `discr` (SCPMI versus SPMI, respectively). However, the difference is negligible, because the compression of PMI values



**(a)** Transfer from SimLex-999 to MEN

**(b)** Transfer from MEN to SimLex-999

**Figure 4.8:** Model transfer between lexical evaluation datasets

should not affect lexical similarity, as we expect by H10. Context distribution (smoothed versus global) and shifting parameter (higher values of $k$ perform better on MEN) are other differences between the parameter selection based on the two datasets.

## 4.3 Transfer of selected models between datasets

We have identified parameters that lead to high correlation scores with human similarity judgements. However, we did not check whether the chosen models overfit. By transferring chosen models across datasets, that is taking parameters that are good on one dataset and applying them on another dataset, we should see whether chosen models overfit.

### 4.3.1 From SimLex-999 to MEN

The models selected using heuristics based on the SimLex-999 dataset perform well on MEN: for all dimensions, the selected models are statistically indistinguishable from the best possible score (Figure 4.8a). The average relative difference with the upper bound is 0.006, or 0.6%.

The Max-based selection is statistically indistinguishable the upper bound for models with dimensionality greater than 5 000. The average relative difference with the upper bound is 0.039. The higher difference is due to overfitting of the low-dimensional models ($D < 5\,000$), where the average relative difference is 0.09.

In this case, heuristic-based selection leads to better performance than the Max-based selection, supporting H1: Max selection does overfit.

### 4.3.2 From MEN to SimLex-999

Heuristics transferred from MEN to SimLex-999 behave less efficiently, they do not always outperform Max selection, though for high-dimensional spaces (5 0000 dimensions and more) the difference is statistically insignificant (Figure 4.8b). The average relative difference is 0.062, which is ten times more than the transition from SimLex-999 to MEN.

| dimensionality | SimLex999 | men | lexical | freq | discr | cds | neg | similarity |
|---:|---:|---:|---:|---|---|---|---:|---|
| 1 000 | 0.347 | 0.682 | 0.892 | 1 | spmi | global | 1.4 | cos |
| 2 000 | 0.361 | 0.722 | 0.936 | logn | scpmi | global | 1.0 | cos |
| 3 000 | 0.357 | 0.737 | 0.940 | logn | scpmi | 1 | 0.7 | cos |
| 5 000 | **0.363** | 0.742 | 0.951 | logn | scpmi | 1 | 0.7 | cos |
| 10 000 | **0.371** | **0.750** | **0.967** | logn | scpmi | 1 | 0.7 | cos |
| 20 000 | **0.381** | **0.761** | **0.987** | logn | scpmi | 0.75 | 0.7 | cos |
| 30 000 | **0.383** | **0.762** | **0.991** | logn | scpmi | 0.75 | 0.7 | cos |
| 40 000 | **0.384** | **0.761** | **0.991** | logn | scpmi | 0.75 | 0.7 | cos |
| 50 000 | **0.385** | **0.760** | **0.992** | logn | scpmi | 0.75 | 0.7 | cos |

**Table 4.7:** Lexical (combined SimLex-999 and MEN) Max selection. For the individual dataset scores, the scores in bold are indistinguishable from the highest score. For SimLex-999, the highest score is 0.385 and the scores above 0.362 are indistinguishable. For MEN, the highest score is 0.762 and the scores above 0.749 are indistinguishable. The highest combined score is 0.992 and the scores above 0.954 are indistinguishable.

Neither Max selection picks the best possible results when transferred from MEN to SimLex-999, the difference with the best scores is statistically insignificant with 3 000 and more dimensions.The average relative difference is lower than with heuristics: 0.042 versus 0.062. This is similar to the transition in other direction.

Max-based selection leads to better performance than the heuristics for MEN, making a case against H1: Max selection does not overfit.

## 4.4 Universal parameter selection for lexical datasets

This section explores whether there are models that behave well across several datasets. It serves two interests. First of all, both datasets measure similarity, so the underlying method of estimating it ideally should not depend on the dataset. Secondly, by compromising between the two datasets, overfitting should be avoided.

We score the models based on the average of the normalised scores over SimLex-999 and MEN:

$$\text{score}_{lexical}(model) = \frac{1}{2} \times \frac{\text{score}_{SimLex-999}(model)}{\max_m \text{score}_{SimLex-999}(m)} + \frac{1}{2} \times \frac{\text{score}_{MEN}(model)}{\max_m \text{score}_{MEN}(m)} \tag{4.3}$$

This is the simplest scoring methods. It treats the datasets equally by giving equal weights to them, without giving any preferences. We normalise the scores to compensate for various magnitudes of representative results between SimLex-999 and MEN.

The performance of the selected models on both datasets and the normalised average is shown in Table 4.7 (Max selection) and Table 4.8 (selection based on heuristics) and in Figure 4.9.

| dimensionality | SimLex999 | men | lexical | freq | discr | cds | neg | similarity |
|---:|---:|---:|---:|---|---|---|---|---|
| 1 000 | 0.334 | 0.681 | 0.875 | logn | scpmi | global | 1 | correlation |
| 2 000 | 0.349 | 0.718 | 0.918 | logn | scpmi | global | 1 | correlation |
| 3 000 | 0.350 | 0.726 | 0.924 | logn | scpmi | global | 1 | correlation |
| 5 000 | 0.353 | 0.733 | 0.933 | logn | scpmi | global | 1 | correlation |
| 10 000 | **0.362** | 0.743 | **0.951** | logn | scpmi | global | 1 | correlation |
| 20 000 | **0.367** | **0.759** | **0.968** | logn | scpmi | global | 2 | correlation |
| 30 000 | **0.372** | **0.761** | **0.976** | logn | scpmi | global | 2 | correlation |
| 40 000 | **0.373** | **0.760** | **0.977** | logn | scpmi | global | 2 | correlation |
| 50 000 | **0.376** | **0.759** | **0.980** | logn | scpmi | global | 2 | correlation |

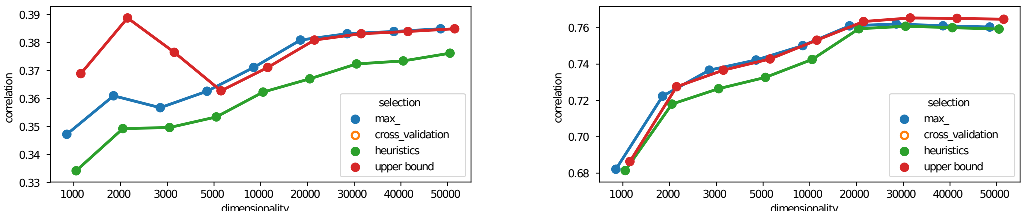
**Table 4.8:** Lexical (combined SimLex-999 and MEN) selection based on heuristics. For the individual dataset scores, the scores in bold are indistinguishable from the highest score. For SimLex-999, the highest score is 0.376 and the scores above 0.353 are indistinguishable. For MEN, the highest score is 0.761 and the scores above 0.748 are indistinguishable. The highest combined score is 0.980 and scores above 0.942 are indistinguishable.

To calculate the minimal required difference for statistical significance (MRDS), the MRDS of the corresponding datasets are put in to formula (4.3). The resulting MRDS value is $\sigma_{0.05}^{0.9} = 0.038$.

### 4.4.1 Max selection

In general, the more dimensions, the better the results are. The selection yields the best results at $D = 50\,000$ for SimLex-999 and at $D = 30\,000$ for MEN. While for SimLex-999, the Max selection is statistically indistinguishable from the best score with 5 000 and more dimensions. For MEN, the performance peaks at 30 000 dimensions and then slightly deviates from the upper bound as the dimensionality increases, however, it is statistically indistinguishable from the highest score with 10 000 and more dimensions.

The Max selection based on the combination of the two lexical datasets is closer to the Max selection based on SimLex-999 (Table 4.1) than on MEN (Table 4.4).



**(a)** SimLex-999 **(b)** MEN

**Figure 4.9:** Performance of models based on the selection over the average lexical performance

### 4.4.2 Heuristics

The linear model achieves an adjusted $R^2$ of 0.817, which is less than the $R^2 = 0.867$ of SimLex-999, but is greater than the $R^2 = 0.733$ of MEN. Table 4.9 shows partial $R^2$s for each parameter—the most influential are similarity, neg and `freq`.

Correlation is the similarity measure of choice (Figure 4.10a). However, the difference between cosine and correlation is not statistically significant for $D < 10\,000$ supporting H9 that correlation is beneficial for high-dimensional spaces.

For the models with dimensionality less than 20 000, shifting should be used with $k = 1$, otherwise, $k = 2$ is preferred (Figure 4.10b). This supports H8 that the more dimensions a model has the sparser it should be.

The choices of $k$ set to 0.2, 0.5 and N/A lead to the scores that are statistically significantly different from the best score. Shifting $k = 0.7$ leads to the statistically different results with $D \geqslant 20,000$; $k = 1$ with $D \geqslant 30\,000$; $k = 1.4$ with $D = 50\,000$; $k = 2$ with $D \leqslant 2\,000$; $k = 5$ with $D \leqslant 10\,00$; and $k = 7$ with $D \leqslant 20$.

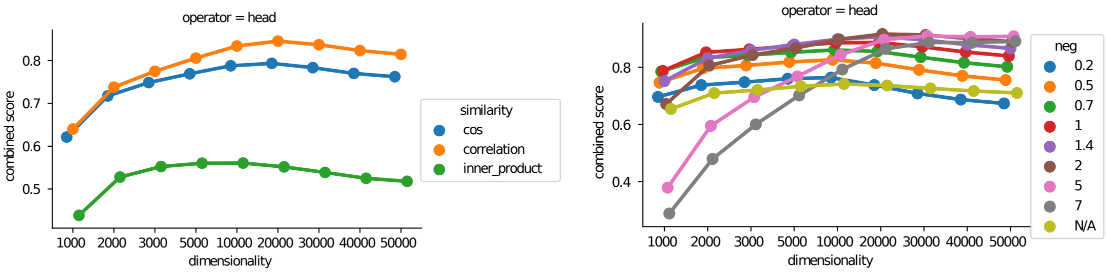
| parameter | partial $R^2$ |
|---|---|
| similarity | 0.299 |
| neg | 0.280 |
| freq | 0.231 |
| dimensionality | 0.095 |
| discr | 0.095 |
| cds | 0.076 |

**Table 4.9:** Lexical feature ablation

The frequency parameter $\log n$, on average, performs the best as the frequency component (Figure 4.11a). But for $D < 30\,000$, 1 performs statistically indistinguishably, supporting H6 that the frequency component is most useful for high-dimensional spaces.

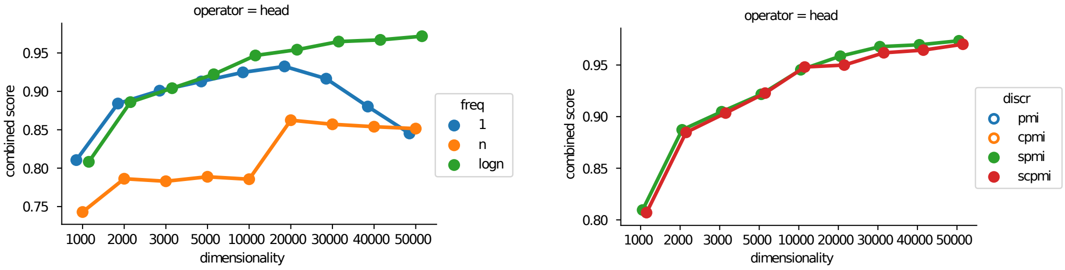
SCPMI is the preferred discrimination component, but SPMI is statistically insignificantly different to it (Figure 4.11b), backing up H10 that PMI value compression is not needed in lexical tasks.

Global context probabilities, on average, behave the best (Figure 4.12). However, global context probabilities and local context probabilities with $\alpha = 1$ yield statistically insignificantly



**(a)** Similarity **(b)** neg

**Figure 4.10:** Lexical influence of similarity and neg



**(a)** Lexical influence of `freq`.

**(b)** Lexical influence of `discr`.

**Figure 4.11:** Lexical influence of `freq` and `discr`

different results for $D > 2\,000$, giving support to H7 that context distribution smoothing is needed in high-dimensional spaces.

### 4.4.3 Comparison with single dataset based selections

Both selection methods mostly agree on frequency ($\log n$) and discriminativeness (SCPMI).

Context probability distribution smoothing varies between the selection methods but follows the corresponding procedures based on MEN.

The Max-based selection for `neg` follows the Max selection on SimLex-999.

Even though the similarity choice is different between the Max-based and heuristic-based selections, it is consistent with SimLex-999 in both cases and with MEN for the heuristic-based selection.

For the Max-based selection, the average difference is 0.020 on SimLex-999 and 0.004 for MEN.

For the heuristics-based selection, the average difference is 0.048 for SimLex-999 and 0.010 for MEN, which is within the 10% limit set by H2.

Max selection behaves better than the heuristics-based selection on the average difference, but we cannot check how well these two selections behave on other lexical datasets. This is

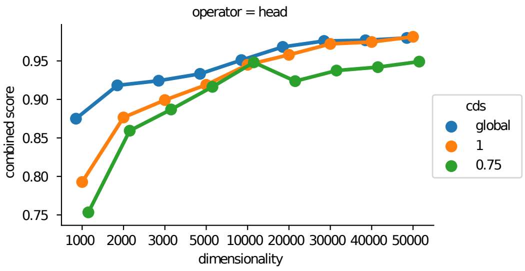


**Figure 4.12:** Lexical influence of `cds`

| Model | SimLex-999 | MEN |
|---|---|---|
| PPMI* | 0.393 | 0.745 |
| SVD* | **0.432** | **0.778** |
| SGNS* | **0.438** | **0.774** |
| GloVe* | 0.398 | 0.729 |
| This work | 0.384 | 0.764 |

**Table 4.10:** Our models in comparison to the previous work on lexical tasks. *Results reported by Levy et al. (2015). Values in bold are statistically indistinguishable from the highest score.

evidence against H1 suggesting that testing on multiple datasets avoids overfitting and manual selection becomes too conservative.

Based on the experiments, lognSCPMI with shifting close to 1 is the quantification of choice for the lexical tasks, however, more work needs to be done to find a robust choice for context distribution smoothing and similarity measure.

## 4.5 Conclusion

Lexical experiments give support to most of the stated hypotheses. The optimal parameter choice depends on dimensionality (H5). In particular, non constant frequency component (H6), context distribution smoothing (H7) and shifting (H8) are recommended to be applied for spaces with $D \geqslant 10\,000$.

The switch at 10 000 dimensions is a "parameter sweet spot," as parameter choice is not significant at these points; the most representative example of this is the behaviour of `cds` on SimLex-999 (Figure 4.3). After that point, performance either converges (supporting H3), as in the case of `neg` on SimLex-999 (Figure 4.2c), or there is one dominant choice, as for `freq` on SimLex-999 (Figure 4.2b).

As expected, we did not see a significant influence of the "compression" of the PMI values (H10).

We could not find supporting evidence for H1, as Max-selected models performed well on transfer and do not overfit. Both selection methods are within the 10% difference margin to the highest result (H2), suggesting that there indeed might be a universal vector space (H4).

On lexical tasks, the best results among the selected models are 0.384 (SimLex-999) and 0.764 (MEN). On the similarity dataset, scores are 0.009 points below the PPMI model of Levy et al. (2015), the difference is not statistically significant. On the relatedness dataset, our score is 0.019 points above Levy et al. (2015), the difference is statistically significant. Note the difference in dimensionality (50 000 in this work, 189 533 in the other work), source corpora (2.8 billion tokens from ukWaC and WaCkypedia in this work, 1.5 billion tokens from an August

2013 Wikipedia dump) and window size (5 versus 2). Table 4.10 shows the results of SVD, SGNS and GloVe-based vector spaces are given for comparison.

The next step is to test compositional models. Before experiments are presented in Chapter 6, a new dataset that provides relevance judgements of pairs of sentences is presented in Chapter 5.

# Chapter 5

# PhraseRel: An IR-inspired compositional dataset

Datasets that quantify relationships between words have a long history.[1] Some of the most well-known datasets are RG65 (Rubenstein and Goodenough 1965), WS353 (Finkelstein et al. 2002), BLESS (Baroni and Lenci 2011), MEN (Bruni et al. 2014) and SimLex-999 (Hill et al. 2015). All of them have been applied to evaluate distributional models of meaning.

Naturally, the idea of capturing lexical relationships was extended to phrases and sentences. Many phrase and sentence datasets exist, such as the MS Paraphrase Corpus (Dolan and Brockett 2005) or the phrase entailment dataset of Baroni et al. (2012). These are potentially applicable to evaluating compositional distributional models. We say potentially because the grammar-preserving compositional settings (Baroni et al. 2014a, Coecke et al. 2010) rely on high-dimensional tensor spaces which do not, at the moment, scale up to the complex syntactic structures of such datasets. Instead, a family of phrasal, mostly *similarity*-oriented datasets (Grefenstette and Sadrzadeh 2011a, Kartsaklis and Sadrzadeh 2013; 2014, Mitchell and Lapata 2008) are widely used for the evaluation of these models, see, for example, Kim et al. (2015a).

We extend the possibility of evaluating compositional distributional models by proposing a task and a corresponding dataset with controlled syntax. The task focuses on *relevance*, a notion from Information Retrieval (IR). Further, this task will assist with the understanding of how compositional distributional models can be used in IR, without limiting the choice of compositional methods. Much of the original research in distributional semantics has stemmed from vector models of IR, but the IR-NLP connection has been less apparent in the newer compositional distributional models; our work attempts to bring them back together.

In a preliminary experiment (Milajevs et al. 2015) we evaluated distributional methods in an

[1] This chapter addresses the evaluation concerns stated in Milajevs et al. (2015), which was presented at ICTIR 2015. The dataset and all referred data files are available at `http://www.eecs.qmul.ac.uk/~dm303/thesis`.

IR-inspired setting. We based the experiment on the sentence similarity dataset KS14. In that dataset, each sentence is paired with three other sentences: a highly similar sentence, a medium similar sentence and a not similar sentence, see below for more details. In that experiment, distributional models improved over an IR baseline.

In that work, we assumed that in the retrieval scenario similarity scores can be converted to retrieval score rankings, in other words, for each query phrase its most similar counterpart is the most relevant document, the somewhat similar counterpart is the second relevant document, and, finally, the dissimilar sentence is the least relevant document.

To drop the assumption that similarity corresponds to relevance, we introduce a dataset that measures relevance. To develop the dataset, we took the KS14 dataset as the base, extensively extended it with query-document phrase pairs and re-annotated it using Amazon Mechanical Turk, asking for relevance judgements.

Due to the novelty of the method used in the creation of the dataset (which makes it measure relevance rather than similarity), the effort dedicated to it and since it has not been published, we devote a separate chapter to explain it.

## 5.1 Preparation of candidate entries for the dataset

As the base, we took the 72 unique sentences from the KS14 dataset of Kartsaklis and Sadrzadeh (2013; 2014).[2] These sentences formed the candidate *query sentences*. We paired each query sentence with 23 *document sentences* obtained by four conceptually different methods. Note that even though similarity is different from relevance, it has been used in the IR setting—see for example Kim et al. (2016).

### 5.1.1 Entries taken from the KS14 dataset

Each query sentence is paired with a counterpart sentence from the high similarity band[3] in the KS14 dataset by treating the similarity band assignments as symmetric. For example, given a KS14 combination

$$(\text{agent}, \text{sell}, \text{property}), (\text{delegate}, \text{buy}, \text{land})$$

in the high similarity band, we generated two query-document permutations listed in Figure 5.1 with the method labelled as `ks14`.

[2] `http://compling.eecs.qmul.ac.uk/wp-content/uploads/2015/07/KS2014.txt`

[3] The KS14 dataset consists of three bands of intended similarity:
- high similarity `emnlp2013_turk_HighSim.txt`,
- medium similarity `emnlp2013_turk_MedSim.txt`,
- low similarity `emnlp2013_turk_LowSim.txt`.

| Query | | | Document | | | Method | Relevance | | | Diverse |
|---|---|---|---|---|---|---|---|---|---|---|
| Subject | Verb | Object | Subject | Verb | Object | | Mean | Std. | Type | |
| delegate | buy | land | agent | sell | property | ks14 | 1.33 | 1.53 | strict | false |
| agent | sell | property | delegate | buy | land | ks14 | 2.00 | 1.00 | strict | false |
| agent | sell | property | representative | exchange | possession | wordnet:hyper | 1.00 | 1.73 | N/A | false |
| agent | sell | property | deputy | trade | estate | wordnet:hypo | 1.00 | 0.00 | loose | false |
| agent | sell | property | people | buy | home | frequency:0 | 1.33 | 1.53 | strict | false |
| agent | sell | property | company | offer | product | frequency:1 | -2.00 | 1.73 | N/A | false |
| agent | sell | property | people | advertise | product | frequency:2 | -2.00 | 1.00 | N/A | false |
| agent | sell | property | family | buy | home | selection | -1.33 | 1.53 | N/A | false |
| agent | sell | property | company | specify | need | selection | -3.00 | 0.00 | N/A | false |
| agent | sell | property | people | represent | set | selection | -3.00 | 0.00 | N/A | false |
| student | acquire | skill | student | gain | experience | selection | 2.00 | 1.00 | strict | true |

**Figure 5.1:** An Example of query-document pairs

### 5.1.2 Entries generated from WordNet

We generated two more document sentences for each query based on *hyponymy* and *hypernymy* relations from WordNet (Miller 1995). For each query sentence, one document sentence was manually generated by substituting the words of a sentence with their *hypernymy* and another document sentence was obtained using *hyponymy*.

During the manual process of retrieving hypernymy and hyponymy, we noticed that some were very good, for instance, *datum*: *information* and *statistics*. However, some candidates were problematic, such as *party*: *political unit* and *communist party* because we were looking for single words rather than phrases. In such cases, we dropped adjectives and adverbs or ignored a candidate. Two instances of WordNet-based sentence generation are shown in Figure 5.1 with the method set to `wordnet:hyper` and `wordnet:hypo`.

### 5.1.3 Entries generated using ukWaC phrase frequencies

To generate more document sentences we used a dependency-parsed version of ukWaC (Ferraresi et al. 2008). We extracted all possible subject-verb-object triplets from the corpus with their frequencies. Then, to generate document sentences for a query, we extracted candidate subjects, verbs and objects independently.

Consider that we need to extract candidate objects for the phrase *agent sell property*. We retrieve objects of the 10 most frequent triplets with the same subject and object. In case there are fewer triplets like this, we retrieve additional subjects of the most frequent triplets that share only the subject or only the verb to fetch 10 subjects in total, giving preference to more frequent triplets. Candidate subjects and verbs are obtained in the same way.

Once candidate subjects, verbs and objects are fetched, we ranked all possible combinations of them by the frequency of appearance in ukWaC and select the top 7 triplets. Documents generated in this manner are labeled as `frequency:*` in Figure 5.1, where the numbers are the frequency ranks of the triplets.

### 5.1.4 Semantically similar entries

To obtain more query-document pairings, we computed the cosine similarity of all queries and documents generated by the model based on the SPMI weighting, $k = 1$, with multiplicative composition. We appended 13 more documents most similar to a corresponding query to each query-document pool. In Figure 5.1 these pairs are labeled as `selection`.

Usage of a similarity metric is justified and does not lead to a circularity because the selected pairs are judged by humans. If a selected entry is considered similar by a distributional model, but not by a human, the task becomes more difficult for distributional models. Moreover, the candidate sentences are likely to have low similarity score (because document sentences do not share same topic), so mostly irrelevant documents are appended.

We had 23 unique documents (1 `ks14`, 2 `wordnet`, 7 `frequency` and 13 `selection`) for each of the 72 queries, making 1656 query-document pairs to be evaluated by humans.

## 5.2 Human evaluation

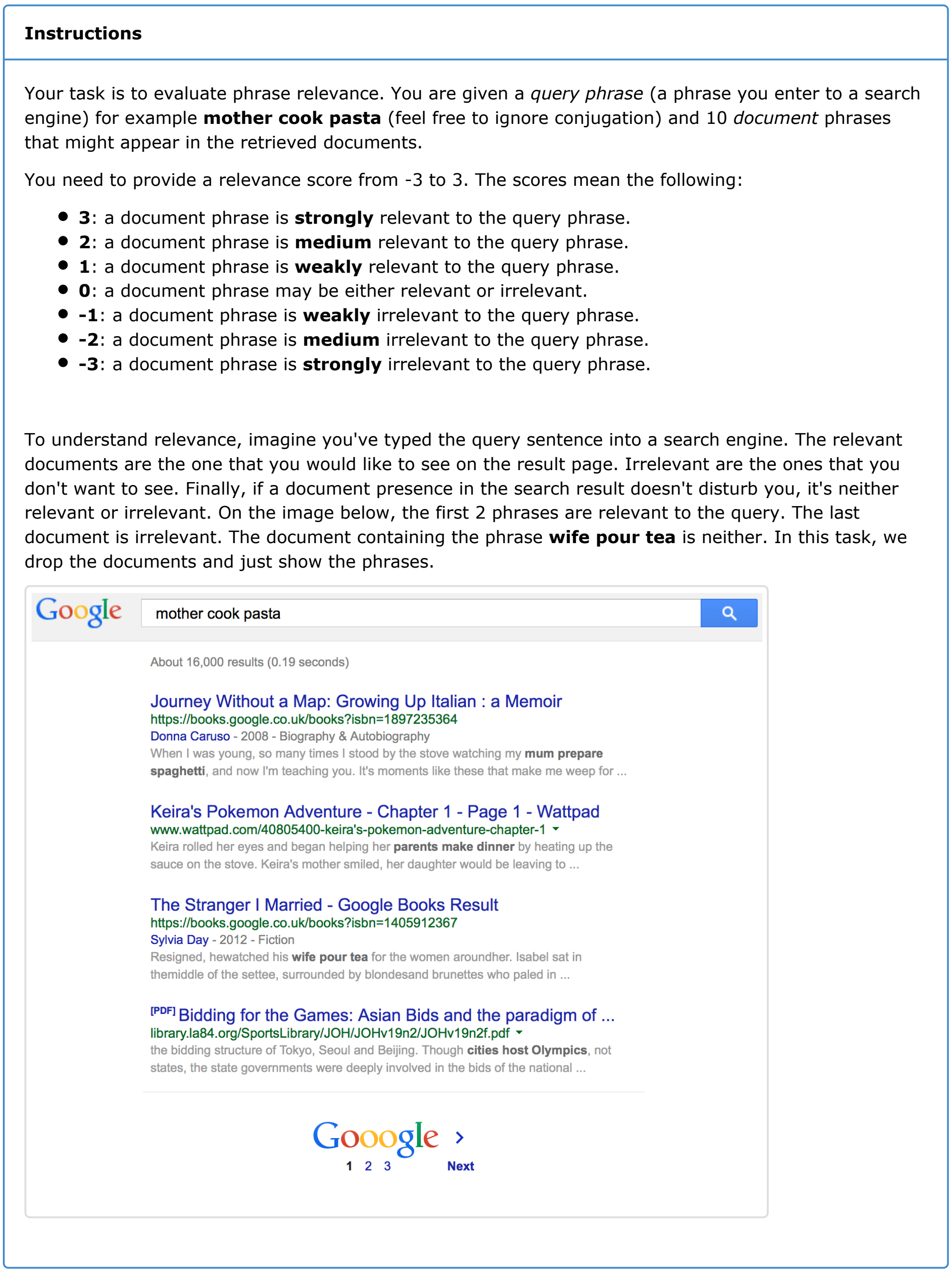
We recruited 19 human subjects using the Amazon Mechanical Turk platform.[4] Before answering a question, the subjects were given the instructions shown in Figure 5.2.

Each query-document pair was judged by three different people. The data was collected in two batches. Each question (or a HIT, Human Intelligence Task, in the Mechanical Turk terminology) consisted of a query phrase and 20 document sentences for the first batch and 10 documents for the second batch. Documents were shuffled before being shown to a human subject. Each document had to be judged for relevance using the following scoring system:

- **-3**: strong irrelevance,
- **-2**: medium irrelevance,
- **-1**: weak irrelevance,
- **0**: a document may be either relevant or irrelevant,
- **1**: weak relevance,
- **2**: medium relevance,
- **3**: strong relevance.

An example question is shown in Figure 5.3. The `phraserel-raw.csv` file consists of human judgements together with anonymized Worker and HIT identifiers.

[4] https://requester.mturk.com/



**Instructions**

Your task is to evaluate phrase relevance. You are given a *query phrase* (a phrase you enter to a search engine) for example **mother cook pasta** (feel free to ignore conjugation) and 10 *document* phrases that might appear in the retrieved documents.

You need to provide a relevance score from -3 to 3. The scores mean the following:

- **3**: a document phrase is **strongly** relevant to the query phrase.
- **2**: a document phrase is **medium** relevant to the query phrase.
- **1**: a document phrase is **weakly** relevant to the query phrase.
- **0**: a document phrase may be either relevant or irrelevant.
- **-1**: a document phrase is **weakly** irrelevant to the query phrase.
- **-2**: a document phrase is **medium** irrelevant to the query phrase.
- **-3**: a document phrase is **strongly** irrelevant to the query phrase.

To understand relevance, imagine you've typed the query sentence into a search engine. The relevant documents are the one that you would like to see on the result page. Irrelevant are the ones that you don't want to see. Finally, if a document presence in the search result doesn't disturb you, it's neither relevant or irrelevant. On the image below, the first 2 phrases are relevant to the query. The last document is irrelevant. The document containing the phrase **wife pour tea** is neither. In this task, we drop the documents and just show the phrases.

About 16,000 results (0.19 seconds)

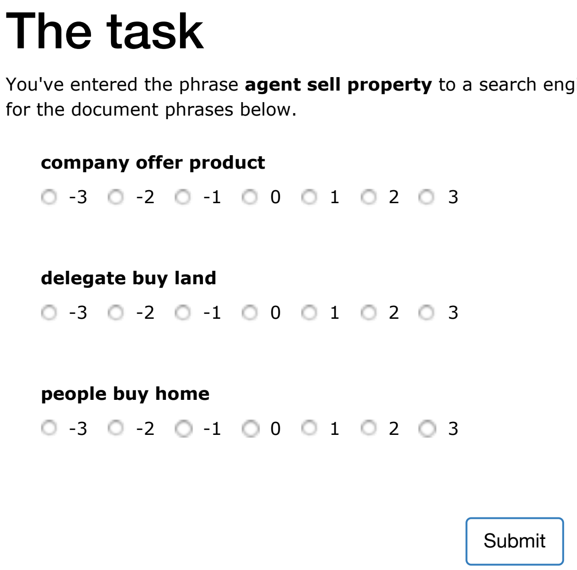
Journey Without a Map: Growing Up Italian : a Memoir
https://books.google.co.uk/books?isbn=1897235364
Donna Caruso - 2008 - Biography & Autobiography
When I was young, so many times I stood by the stove watching my **mum prepare spaghetti**, and now I'm teaching you. It's moments like these that make me weep for ...

Keira's Pokemon Adventure - Chapter 1 - Page 1 - Wattpad
www.wattpad.com/40805400-keira's-pokemon-adventure-chapter-1
Keira rolled her eyes and began helping her **parents make dinner** by heating up the sauce on the stove. Keira's mother smiled, her daughter would be leaving to ...

The Stranger I Married - Google Books Result
https://books.google.co.uk/books?isbn=1405912367
Sylvia Day - 2012 - Fiction
Resigned, hewatched his **wife pour tea** for the women aroundher. Isabel sat in themiddle of the settee, surrounded by blondesand brunettes who paled in ...

[PDF] Bidding for the Games: Asian Bids and the paradigm of ...
library.la84.org/SportsLibrary/JOH/JOHv19n2/JOHv19n2f.pdf
the bidding structure of Tokyo, Seoul and Beijing. Though **cities host Olympics**, not states, the state governments were deeply involved in the bids of the national ...

**Figure 5.2:** Instructions given to the Mechanical Turkers



The task

You've entered the phrase **agent sell property** to a search engine such as Google. Give the relevance scores for the document phrases below.

**company offer product**

○ -3 ○ -2 ○ -1 ○ 0 ○ 1 ○ 2 ○ 3

**delegate buy land**

○ -3 ○ -2 ○ -1 ○ 0 ○ 1 ○ 2 ○ 3

**people buy home**

○ -3 ○ -2 ○ -1 ○ 0 ○ 1 ○ 2 ○ 3

**Figure 5.3:** A sample question

## 5.3 Final dataset construction

### 5.3.1 Pair classification

The pairs have been classified by the individual relevance judgements. A pair is of the *strict relevance* type if all human judgements are greater than or equal to 0 and there is at least one 3 (strong relevance). A pair of the *loosely relevant* type is a pair where all human judgements are greater than or equal to 0. Figure 5.1 shows the mean and standard deviation of human judgements per query-document pair and the relevance division of a pair if applicable. There are 110 (or 7%) strictly relevant pairs and 221 (13%) loosely relevant pairs.

### 5.3.2 Query classification

Each query has been classified by the number of strictly relevant documents it has been paired with. On average, each query in the dataset is paired with 3.3 loosely relevant and 1.6 strictly relevant documents. The minimum number of loosely relevant documents per query is 1, the maximum is 14. Regarding the strict relevance, there are 8 queries without a single strictly relevant document, and the maximum number of strictly relevant documents per query is 7.

A query is *diverse* if it is paired with 4 or more loosely relevant documents. There are 28 (42%) such queries. On average, diverse queries are paired with 5.2 loosely relevant documents and 2.4 strictly relevant documents. The 28 queries form $28 \times 23 = 644$ query-document pairs that form the dataset.

### 5.3.3 Evaluation and statistical significance testing

Precision at 3 is the evaluation metric suggested for this dataset, it measures for which proportion of queries there is at least one relevant document among the top three ranked query-document pairs.

To evaluate statistical significance between two results a sign test should be used.

The difference between two results sill indicate the minimal number of improvement. Consider a dataset of 10 entries and a result difference of 0.3. This means that there were either 3 positive differences and 0 negative difference, or 4 positive differences and 1 negative difference, 5 positive differences and 2 negative differences and 6 positive differences and 3 negative differences (the difference between the positive and negative differences must be 3, but the sum less or equal to 10). For the difference of 0.3 to satisfy the minimum difference for statistical significance all the possibilities must be statistically significant, the sign test must return $p$ value less then the threshold. For phraserel, $\sigma_{0.05} = 0.43$.

## 5.4 Conclusion

This chapter introduced a new dataset that provides relevance judgements of pairs of sentences. Chapter 6 reports experiments on compositional datasets that capture sentence similarity and relatedness.

# Chapter 6

# Relationships between sentences

In this chapter, we study the relationship between lexical representations and various methods of composition and look for the optimal lexical representations for additive, multiplicative and Kronecker-based compositional methods.[1] We do not make any assumption regarding lexical representations and test all of them to see their behavioural patterns in compositional setting.

Here we report the results on the compositional datasets KS14 (Kartsaklis and Sadrzadeh 2014), GS11 (Grefenstette and Sadrzadeh 2011b) and PhraseRel (Chapter 5). For the first two datasets, the evaluation is done similarly to the lexical evaluation (Chapter 4) by computing Sperman-$\rho$ rank correlation between human judgements and model estimation. If a dataset provides several human judgements for a sentence pair, the judgements are averaged before computing the correlation. For PhraseRel, we report relevant@3, the measure that is the proportion of sentences for which the top-3 most similar neighbours contain at least one sentence that was judged relevant with respect to the source sentence.

We show that the optimal choice of lexical parameters depends on the method of composition, so, for example, with addition the vectors should be sparser than the vectors that are used with multiplication and Kronecker. We also show that the parameter choice that is optimal for lexical tasks is sub-optimal for compositional tasks especially for multiplication and Kronecker.

## 6.1 Experiments on KS14 dataset

KS14 is a sentence similarity dataset prepared by Kartsaklis and Sadrzadeh (2014). It consists of pairs of transitive sentences that are judged by similarity. The goal is to achieve high corre-

[1] This chapter includes the results presented in Milajevs et al. (2014) at EMNLP 2014.

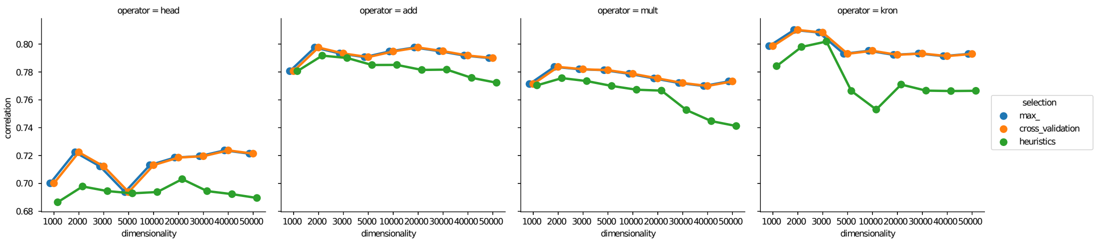


**Figure 6.1:** KS14 results

lation with human judgements in predicting sentential similarity. We also report behaviour of a baseline operator `head`, which ignores the subject and the object of a sentence and makes the vector of a sentence equal to the vector of the head word, in our case the verb. The minimum significant difference for KS14 is $\sigma^{0.9}_{0.05} = 0.07$.

### 6.1.1 Max selection

Figure 6.1 shows the performance of compositional models on the sentence similarity dataset KS14. All operators outperform the non-compositional `head` operator, addition and Kronecker statistically significantly outperform `head` operator. Table A.1 shows the performance of models selected by Max selection together with the selected parameters.

Kronecker with a few thousand dimensions and correlation as the similarity measure gives the highest scores, supporting H12 that word order is important in predicting similarity. As the dimensionality increases, Kronecker performance stays constant. Addition is slightly better than multiplication, but the performance of both peaks at 2 000 dimensions and decreases as dimensionality increases. There is no statistically significant difference between the highest score of Kronecker and the highest scores of addition and multiplication.

Parameters of the baseline compositional method `head` are similar to the lexical Max selection (Table 4.7), with an exception of `neg`, which controls vector sparsity, where higher values that are similar to MEN (Table 4.4) are chosen.

All compositional operators agree in the choice of `freq` ($\log n$), `discr` (SCPMI) and similarity (correlation—note that Kronecker was tested only with the inner product for $D > 3\,000$ because of limited computational resources).

Compositional operators perform best with constant `freq` of 1, in contrast to the lexical setting, where $\log n$ is more beneficial. This might be because during composition the $\log n$ term dominates over the PMI value and minimises its effect.

Local context probabilities perform better in compositional tasks than in lexical tasks. Multiplication benefits from the unsmoothed distribution probability, while high-dimensional models perform best with smoothing ($\alpha = 0.75$), supporting H7 that context smoothing is needed only for high-dimensional models. The only exceptions are additive models with $D < 5\,000$, where global probabilities perform best.

For low-dimensional spaces, addition performs best with sparse spaces ($k > 1, D < 5\,000$), but for high-dimensional spaces, addition performs best with dense spaces ($k = 0,7, D \geqslant 5\,000$). This is against H8 that states the opposite.

Multiplication, independently of dimensionality, performs best with dense spaces ($k = 0.2$).

Kronecker—in contrast to addition—performs best with dense low-dimensional models ($k = 0.2$, $D < 5\,000$) and sparser high-dimensional models ($k = 0.7$, $D \geqslant 5\,000$), which complies with H8. However, this difference might be explained by the change of the similarity measure, which is the inner product for $D \geqslant 5\,000$.

### 6.1.2 Heuristics

The linear model achieves an $R^2$ of 0.79. The partial $R^2$s are shown in Table 6.1. The most influential parameters are neg, freq, compositional operator and cds. Interestingly, similarity has much less influence on this compositional dataset than on lexical datasets, where for Sim-Lex-999 (Table 4.3) and combined (Table 4.9) it is the most influential parameter. Also, note that dimensionality has the lowest partial $R^2$.

| parameter | partial $R^2$ |
|---|---|
| neg | 0.33 |
| freq | 0.31 |
| operator | 0.30 |
| cds | 0.14 |
| similarity | 0.06 |
| discr | 0.05 |
| dimensionality | 0.03 |

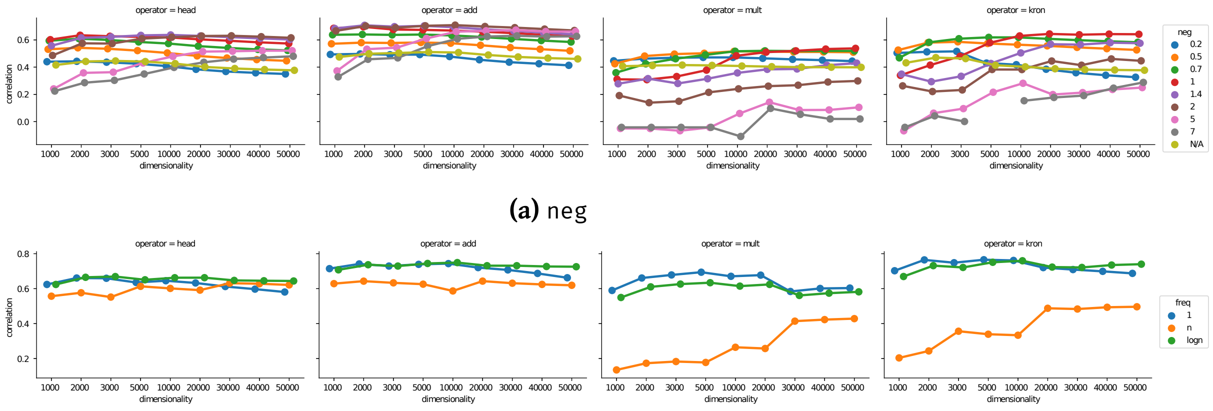
**Table 6.1:** KS14 feature ablation

#### Shifting

For the baseline operator head, the best shifting choice of $k$ is 1 for spaces with dimensionality less than 5 000 (Figure 6.2a); $k = 0.7$, $k = 1,4$ and $k = 2$ are statistically indistinguishable from the best score. For $5\,000 \leqslant D < 30\,000$, head behaves best with $k = 1.4$; $k = 0.7, 5\,000 \leqslant D \leqslant 10\,000$ is statistically insignificant from the best score, as well as $k = 1$ and $k = 2$. For $D \geqslant 3\,000$, $k$ should be set to 2; but $k = 1$ and $k = 1.4$ are statistically indistinguishable from it.

For addition, spaces with $D < 20\,000$ should be used with $k = 1.4$; $k = 0.7, D \leqslant 5\,000$ is statistically indistinguishable from the best score, as well as $k = 1$ and $k = 2$. For $D \geqslant 20\,000$, $k = 2$ should be used, however $k = 1$, $k = 1.4$, $k = 5$ and $k = 7, D \geqslant 30\,000$ are statistically indistinguishable from it.

For multiplication, there are three most beneficial choices: for $D < 10\,000$ $k = 0.5$ ($k = 0.2$ and $k = 0.7$ are statistically indistinguishable), for $10\,000 \leqslant D < 30\,000$ $k = 0.7$ ($k = 0.2$, $k = 0.5$ and $k = 1$ are statistically indistinguishable) and, finally, for $D > 30\,000$ $k = 1$ ($k = 0.5$ and $k = 0.7$ are statistically indistinguishable).



**(a)** `neg`

**(b)** `freq`

**Figure 6.2:** KS14 influence of `neg` and `freq`

Kronecker shows a behaviour similar to multiplication for $k$ as dimensionality increases, but prefers sparser spaces. For $D < 3\,000$: $k = 0.5$ ($k = 0.2$ and $k = 0.7$ are statistically indistinguishable), for $3\,000 \leqslant D < 20\,000$: $k = 0.7$ ($k = 0.5$ and $k = 1$ are statistically indistinguishable) and for $20\,000 \leqslant D$: $k = 1$ ($k = 0.7$ and $k = 1.4$ are statistically indistinguishable).

All operators behave in accordance to H8 that low-dimensional spaces benefit from being dense, while high-dimensional spaces benefit from being sparse.

## Frequency

The best option of frequency for the baseline operator `head` is $\log n$ (Figure 6.2b). The constant frequency 1 is statistically indistinguishable from $\log n$, but its performance declines for spaces with $D > 20\,000$. The linear choice $n$ gives statistically different results for $D$ set to 2 000, 3 000, and 20 000.

For addition, frequency should be set to 1 for spaces with $D < 5\,000$ and to $\log n$ otherwise. However, there is no statistically significance difference between 1 and $\log n$. There is statistically significant difference for $n$.

There is one choice of frequency for multiplication: 1. However, there is no statistically significant difference for $\log n$ and 1. There is statistically significant difference for $n$.

Kronecker follows addition with regard to frequency, but the split point is $D = 10\,000$: low-dimensional spaces should be used with constant frequency 1, and high-dimensional spaces with $\log n$, however there is no statistically significance difference. Again, $n$ gives statistically significantly lower results.

H6—that non-constant frequency is beneficial for high-dimensional spaces—is supported by all operators in this dataset looking to the highest score, however the difference is not statistically significant.

### Context distribution smoothing

The baseline operator `head` with spaces with dimensionality less than 20 000 should be used with global probabilities, and more dimensional models should be used with smoothed, local probabilities: $\alpha = 0.75$ (Figure 6.3a). However, there is not statistically significant difference between the smoothing choices.

All other operators perform best with global context probability. Even though local context probability with $\alpha = 1$ is with very few exceptions is statistically indistinguishable, H7 is not supported by this dataset: context distribution smoothing does not affect high-dimensional spaces.

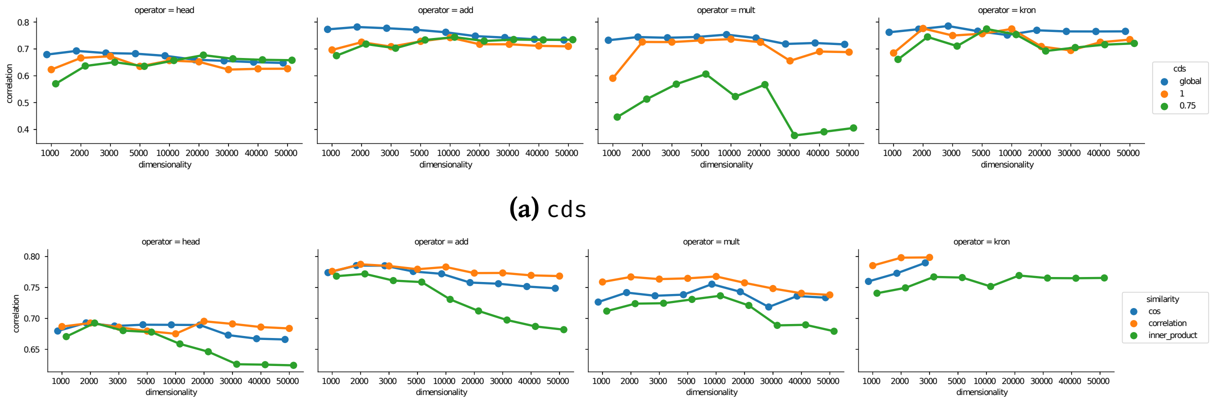
### Similarity

The baseline operator `head` on spaces with $D < 20\,000$ performs best with cosine similarity, while more dimensional models prefer correlation as the similarity measure (Figure 6.3b), however there is no statistically significant difference between the similarity measures.

Other operators work best with correlation, however there is no statistically significant difference between the operators also in this case. In the case of multiplication, correlation dominates over cosine, even for small values of $D$. There is little to say about Kronecker, as it is tested only with the inner product for spaces with $D > 3\,000$, due to its computational complexity ($\mathcal{O}(n^2)$ with respect to the number of vector components).

### Discriminativeness

The baseline operator `head` with $D < 20\,000$ prefers SCPMI as the discriminativeness weighting. SPMI is preferred otherwise (Figure 6.4). However, there is no statistically significant difference between the SCPMI and SPMI.



**(a)** `cds`

**(b)** `similarity`

**Figure 6.3:** KS14 influence of `cds` and `similarity`

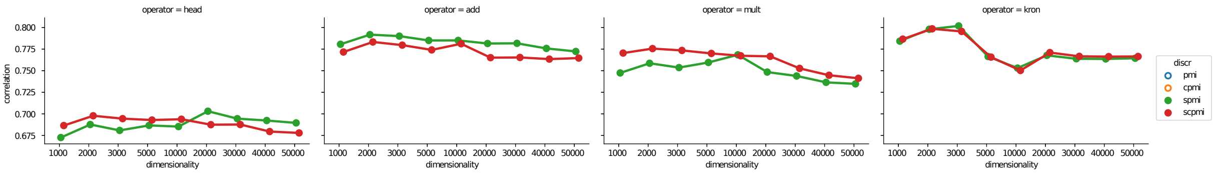


**Figure 6.4:** KS14 influence of `discr`

For addition, SPMI is the better choice, but again without a statistically significant difference. For multiplication, SCPMI gives higher score, but the difference with SPMI is not statistically significant, as expected by H13: the compression of PMI values improves the performance of compositional models.

For Kronecker, the two choices are also statistically indistinguishable. For spaces with dimensionality less than 20 000, SPMI is slightly better; for spaces with greater dimensionality—SCPMI.

### 6.1.3 Difference between Max selection and heuristics on KS14

Table A.2 shows the selection based on heuristics, which is more homogeneous than the selection based on the highest score (Table A.1). Both methods agree on the similarity choice (with the exception of `head`).

Multiplication agrees on the majority of parameters, except `cds` and `neg`. Local probabilities ($\alpha = 1$) and $k = 0.2$ give the highest score, while manual selection picked global context probabilities with `neg` in the range of 0.5 to 1, in accordance with H8 that high-dimensional spaces should be sparser.

The average relative difference between Max and heuristic-based selections is 0.02. Per operator, the differences are: 0.03 (`head`), 0.01 (addition), 0.02 (multiplication) and 0.03 (Kronecker). All values are within the 10% set by H2. The differences in the scores are not statistically significant.

## 6.2 Experiments on GS11 dataset

GS11 is a dataset of transitive sentences (Grefenstette and Sadrzadeh 2011a;b). It consists of ambiguous transitive verbs together with their arguments and two landmark verbs that each disambiguate a particular sense of the ambiguous verb. Human judgements provide pairwise similarity scores between the sentence with the ambiguous verb and the two sentences with the landmark verbs. The minimum significant difference for KS14 is $\sigma^{0.9}_{0.05} = 0.05$.

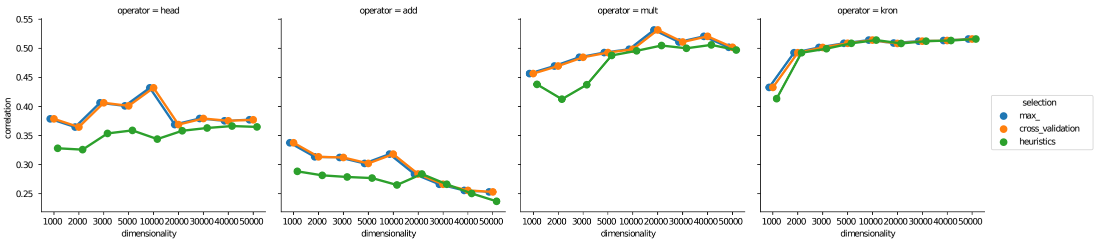


**Figure 6.5:** GS11 results

### 6.2.1 Max selection

Figure 6.5 shows performance of the compositional models on the verb disambiguation task. Table A.3 shows the selected model performance together with chosen parameters.

Multiplication with 20 000 dimensions gives the highest result of 0.53. Kronecker's score is not statistically significantly different: 0.52 with $D = 50\,000$, giving no support to H12 that word order is important for similarity measurement. Addition statistically significantly worse than the baseline head operator: addition scores 0.34, while head's best performance is 0.43, both of them are statistically significantly different from addition and multiplication.

The behaviour of the baseline operator head is unstable for dimensions less than 20 000, and its best behaviour might be the case of overfitting similarly with SimLex-999. However, models with dimensions greater than 20 000 yield similar scores, even though the parameters are different.

In general, Max parameter selection is very different than the one based on KS14 (Table A.1). Compositional operators behave best with $\log n$ frequency, especially Kronecker. PMI often outperforms other discriminativeness components in the case of head and addition. Global context probability estimation behaves better than local and correlation is not always the best similarity measure.

Addition's behaviour degrades as dimensionality increases, whereas multiplication's behaviour increases, but becomes unstable for spaces with more than 20 000 dimensions. Kronecker depends on the dimensionality the least.

Addition works best with dense models. Multiplication and Kronecker prefer dense, low-dimensional spaces and sparse, high-dimensional spaces, supporting H8 that a frequency component is required for high-dimensional spaces.

### 6.2.2 Heuristics

The linear model achieves an $R^2$ of 0.75. The partial $R^2$ scores are shown in Table 6.2. The most influential parameters are a compositional operator, freq and neg. This is the same as in the case of KS14, but in reverse order (Table 6.1).

### Frequency

On average, $\log n$ behaves best for all operators (Figure 6.6a). For the baseline operator head, the difference between $n$ and the best score becomes significant for $D \geqslant 10\,000$. For addition, constant frequency becomes statistically significantly different for $D \geqslant 20\,000$ and for linear frequency the difference is statistically significant for $D \geqslant 40\,000$. For multiplication, $n$ is statistically significantly different for all dimensions and with constant frequency the difference is statistically significant for $D \geqslant 40\,000$. It is similar for Kronecker: linear frequency is statistically significantly different for $D \geqslant 40\,000$ and for all dimensions with linear frequency.

| parameter | partial $R^2$ |
|---|---|
| operator | 0.37 |
| freq | 0.21 |
| neg | 0.18 |
| similarity | 0.09 |
| cds | 0.05 |
| discr | 0.04 |
| dimensionality | 0.04 |

**Table 6.2:** GS11 feature ablation

The fact that 1 becomes significantly different for addition, multiplication and Kronecker supports H6 that non-constant frequency component is required for high-dimensional spaces.

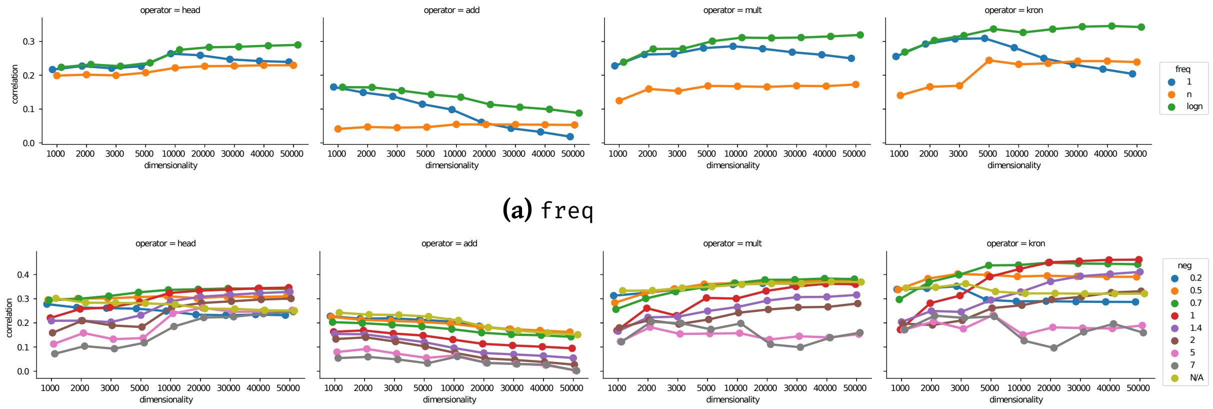
### Shifting

The baseline operator head on average works best with shifted models. For models with dimensionality less than 3 000, $k = 0.5$ is best (values of $k \leqslant 1$ and unshifted models are not statistically significantly different). Otherwise $k = 0.7$ is more beneficial (Figure 6.6b). Here, $k = 0.5$ and $k = 1.0$ are not statistically significantly different for all dimensions; $k = 1.4$ is not statistically significantly different for $D \geqslant 10\,000$, $k = 2.0$ is not statistically significantly different for $D \geqslant 40\,000$.

For addition, models without shifting behave best for $D < 20\,000$, for more dimensional spaces, $k = 0.2$ should be preferred. However, there is no statistically significant difference between $k = 0.2$, $k = 0.5$, $k = 0.7$ and unshifted models. This is a weak support of H8 (high-dimensional vectors should be sparse) because unshifted spaces can be seen as shifted with a very small $\alpha$ value.

Multiplication also works best with unshifted low-dimensional spaces ($D < 5\,000$) and with $k = 0.7$ for high-dimensional spaces. The choices of $k = 0.2$ and $k = 0.5$ are statistically indistinguishable, there is no statistically significance difference with the highest score and $k = 0.7$ and $D \geqslant 2\,000$, as well as with $k = 1$ and $D \geqslant 20\,000$. This supports H8.

Kronecker prefers shifting. For spaces with dimensionality less than 20 000 $k = 0.7$ and $k = 1$ otherwise (however, $k = 1$ is statistically indistinguishable). This is inline with H8. The choice of $k = 0.2$ is statistically indistinguishable for $D \leqslant 2\,000$, the choice of $k = 0.5$ is statistically



**(a)** `freq`

**(b)** `neg`

**Figure 6.6:** GS11 influence of `freq` and `neg`

indistinguishable for $D \leqslant 10\,000$, the choice of $k = 1$ is statistically indistinguishable for $D \geqslant 5\,000$ and unshifted model with $k \leqslant 3\,000$.

## Similarity

The baseline operator `head` and multiplication work best with cosine similarity, addition with correlation and Kronecker with inner product (Figure 6.7a). For the baseline operator and Kronecker, there is no statistically significant difference between similarity measures with an exception of the inner product and $D \leqslant 2\,000$. For multiplication, the inner product is statistically significantly different than both cosine and correlation.

Addition strictly supports H9 that cosine is optimal for high-dimensional spaces: for $D \geqslant 20\,000$ the difference between correlation and cosine is statistically significant, while multiplication supports it by behaving similarly to cosine and correlation.

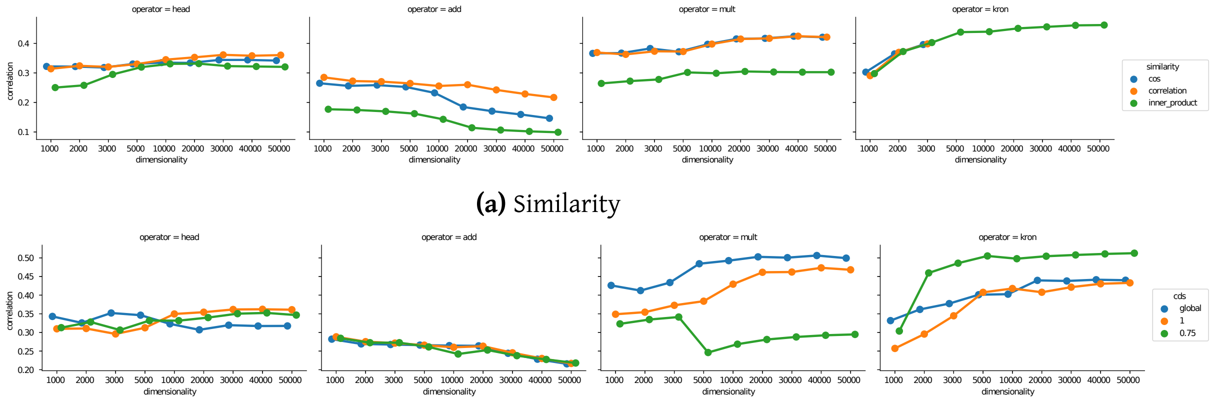
## Context distribution smoothing

The baseline operator `head` with $D < 10\,000$ works best with global context probabilities. For more dimensional spaces, local context probabilities $\alpha = 1$ should be preferred (Figure 6.7b). However the difference between the smoothing choice is not statistically significant.

Addition works best with local probabilities. In the low-dimensional case, when $D < 20\,000$, unsmoothed estimation ($\alpha = 1$) is preferred and $\alpha = 0.75$ should be chosen otherwise. Again, the difference is not statistically significant.

Multiplication works best with global context probabilities. However, with $D \geqslant 20\,000$ the choice of $\alpha = 1$ is not statistically significantly different.

Kronecker works best with smoothed local smoothing ($\alpha = 0.75$) and other choices are statistically significantly different.



**(a)** Similarity

**(b)** `cds`

**Figure 6.7:** GS11 influence of similarity and `cds`

There is no support of H7 (context distribution smoothing leads to optimal results with high-dimensional models) because global context probabilities outperform other choices for multiplication, addition behaves the same with all options and Kronecker works best with $\alpha = 0.75$.

### Discriminativeness

The baseline operator `head` works best with SPMI, but SCPMI is very close and there is no statistically significant difference between discriminativeness choices. (Figure 6.8). Addition works best with PMI for $D < 20\,000$ and SCPMI otherwise, again all discriminativeness choices are statistically similar.

Multiplication is similar to addition in that it prefers PMI in the low-dimensional case and SCPMI in the high-dimensional case, but the change happens at 5 000 dimensions rather than 20 000 however the difference is not statistically significant.

Kronecker with less than 5 000 dimensions prefers SCPMI and SPMI, which is opposite to addition and multiplication, again the difference is not statistically significant.

This dataset does not give evidence to support H13—PMI values should be compressed when used in compositional setting—because there is almost no difference between *PMI and *CPMI.

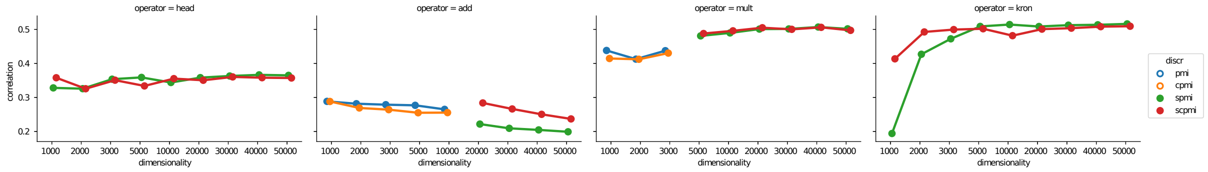


**Figure 6.8:** GS11 `discr`

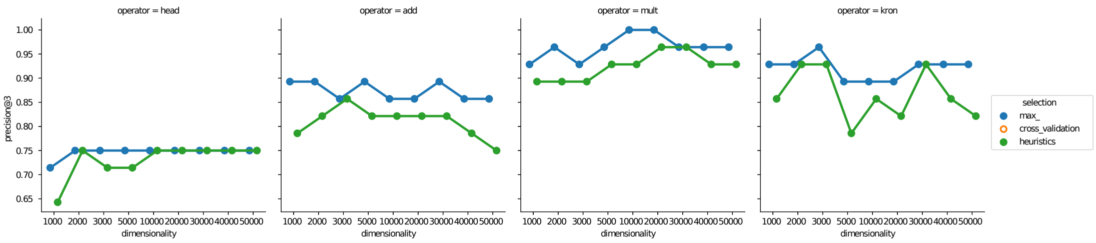


**Figure 6.9:** PhraseRel results

### 6.2.3 Difference between Max selection and heuristics on GS11

Only logarithmic frequency component ($\log n$) was chosen by heuristics (Table A.4), while there is a mix of 1 and $\log n$ in the Max selection (Table A.3).

Kronecker and most of multiplication's discriminativeness choices agree, while for head and addition there is little agreement between parameter selection. The same goes for context distribution smoothing and shifting.

Similarity choice is the same for Kronecker and addition, but head and multiplication—according to heuristics—should be used with cosine similarity, while there is no single metric that leads to maximum performance.

The overall average normalised difference in results between Max and heuristic-based selections is 0.05. Per operator, the differences are: 0.09 head, 0.08 addition, 0.04 multiplication and 0.01 Kronecker. All the values are within the 10% boundary set by H2.

## 6.3 Experiments on PhraseRel dataset

PhraseRel is a dataset that is built for evaluation of distributional models, but instead of similarity judgements, relevance judgements are provided. The evaluation measure is relevance@3. This is the proportion of the retrieval results for which there is a relevant document among the top three ranked documents.

The minimum significance difference for PhraseRel is $\sigma_{0.05} = 0.43$. With such a large value most of the experiments reported below do not produce statistically significant difference, an exception being shifting for multiplication and Kronecker.

### 6.3.1 Max selection

Figure 6.9 shows the performance of the models on the PhraseRel dataset. All operators outperform the non-compositional head baseline. Table A.5 shows the models that yield the best result together with model parameters.

Multiplication, in general, outperforms all other operators, and with the dimensions of 10 000 and 20 000, gets the perfect score of 1.[2] The fact that Kronecker (a word order sensitive operator) is outperformed by multiplication (a word order insensitive operator) gives no support of H12 that word order matters. Model performance weakly depends on the dimensionality for all operators. Multiplication and Kronecker show strong performance similarly to the preliminary study, where similarity was assumed to correspond to relevance (Milajevs et al. 2015).

Addition and Kronecker achieve the best score with constant frequency, head works best with linear frequency and multiplication with sublinear ($\log n$) frequency.

SPMI is the preferred discriminativeness component for the low-dimensional spaces ($D < 10\,000$) for the baseline head, otherwise, SCPMI is the best behaving discr. For addition and the spaces with $D > 1\,000$, SPMI is the best, while for the spaces with the same dimensionality, multiplication prefers CPMI, which is in line with H13: PMI values should be compressed when used in compositional tasks. Kronecker, most of the time, prefers SCPMI.

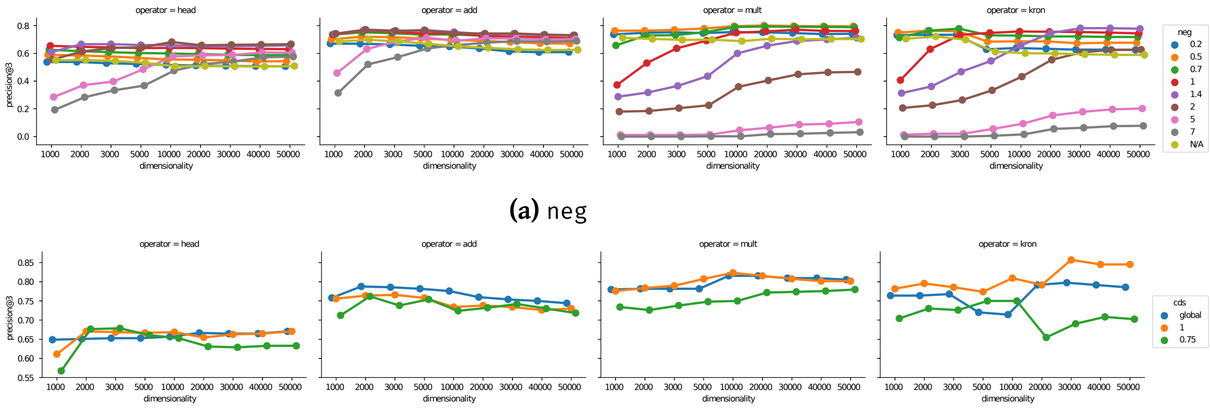
The baseline operator head with dimensions less than 20 000 works best with local smoothed context probabilities, however, for more dimensional spaces, global context probabilities are more competitive. On the contrary, addition prefers smoothed local context probabilities for spaces with dimensions more than 5 000. Multiplication exhibits different pattern: when a model contains few dimensions, it prefers local, smoothed context probabilities, and for high-dimensional spaces it prefers local, but unsmoothed context probabilities, which goes against H7 (context distribution smoothing should be optimal with high-dimensional spaces). Kronecker is inconsistent with regards to the choice of cds, but for models with $D \geqslant 30\,000$, global context probabilities perform the best.

Regarding shifting, the baseline operator head prefers sparse spaces $k > 1$, but as dimensionality increases, the optimal $k$ values decrease. Addition does not show a consistent behaviour with regard to this parameter. Multiplication, in general, benefits from dense, unshifted spaces. Kronecker works best with sparse spaces with increasing sparsity as the dimensionality increases, supporting H8: low-dimensional models should be dense, but high-dimensional models should be sparse.

The baseline operator head benefits from the correlation as the similarity measure, as does multiplication. Addition works best with correlation with spaces $D < 10\,000$, and with the inner product for more dimensional spaces. Multiplication works best with correlation. Kronecker, for spaces with less than 5 000 dimensions, works best with correlation and with the inner product, otherwise.

### 6.3.2 Heuristics

[2] The perfect score of 1 does not mean that the model is perfect, it means that for all queries at least one relevant document was among the top three ranked.



(a) neg

(b) cds

**Figure 6.10:** PhraseRel influence of neg and cds

The linear model achieves an $R^2$ of 0.82. The partial $R^2$ scores are shown in Table 6.3. The most influential parameters are neg, operator and cds, but the first two have partial $R^2$ scores much higher than the other parameters. Table A.6 shows the performance of the chosen models.

| parameter | partial $R^2$ |
|---|---|
| neg | 0.58 |
| operator | 0.35 |
| cds | 0.08 |
| freq | 0.04 |
| similarity | 0.03 |
| dimensionality | 0.03 |
| discr | 0.02 |

**Table 6.3:** PhraseRel feature ablation

### Shifting

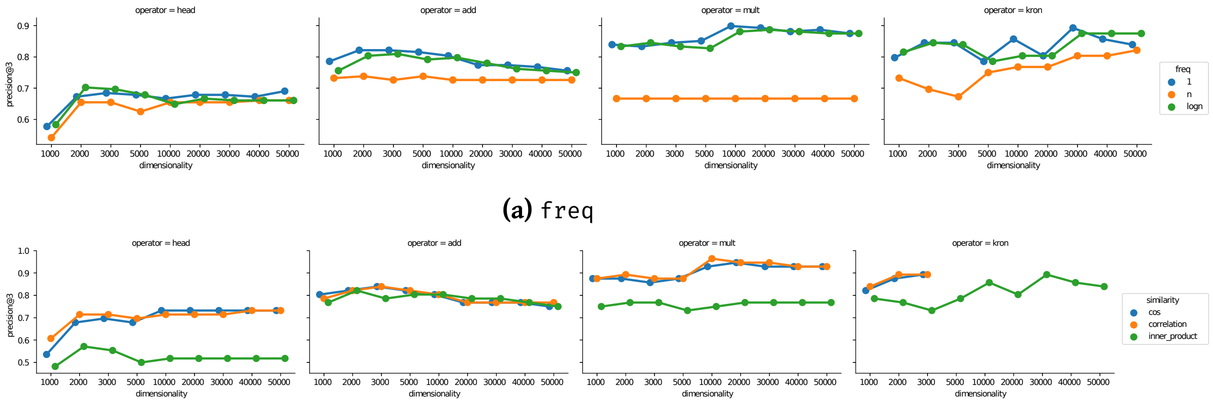
The baseline operator head should be used with $k = 1.4$, addition should be used with $k = 2$ and multiplication should be used with $k = 0.5$ (Figure 6.10a).

Kronecker has three optimal values of $k$ that are proportional to dimensionality. For models with dimensionality less than 5 000, $k = 0.5$ is preferred; for $5\,000 \leqslant D < 20\,000$, the most beneficial choice of neg is $k = 1$; finally, for spaces with more than 20 000 dimensions, $k$ should be set to 1.4. This supports H8 that model sparsity should increase as dimensionality increases.

The only statistically significant difference is for multiplication between $k \in 5, 7$ and the best result and for Kronecker between $k = 7$ for all dimensions and $k = 5$ when $D \leqslant 5\,000$.

### Context distribution smoothing

The best choice for the baseline operator head depends on dimensionality: spaces with less than 10 000 dimensions benefit from smoothed local context probabilities ($\alpha = 0.75$, Figure 6.10b). Addition and multiplication work best with global context probabilities, while Kronecker prefers unsmoothed local probabilities ($\alpha = 1$).



**(a)** `freq`

**(b)** similarity

**Figure 6.11:** PhraseRel influence of `freq` and similarity

### Frequency

The baseline operator`head` works best with linear frequency, but the difference between other options is small (Figure 6.11a). Addition benefits from linear frequency, but sublinear frequency is very close. Multiplication works best with sublinear frequency, but linear is very close to it. Finally, Kronecker works best with $\log n$ with spaces with dimensionality less than 5 000, and with linear frequency with more dimensional spaces.

In general, H6 (non-linear frequency should be used with high-dimensional spaces) holds, because there is no difference between 1 and $\log n$ choices in model performance.

### Similarity

For all operators, there is little difference between cosine and correlation, weakly supporting H9 that correlation should be used with high-dimensional spaces and cosine with low-dimensional spaces.

The baseline operator `head` works best with correlation as the similarity measure with models with $D < 5\,000$, and with cosine for more dimensional ones (Figure 6.11b). Note, however, that the difference between the two is very small.

Addition benefits from cosine when $D < 20\,000$ and from inner product otherwise. But, in the case of addition, all three similarity measures are close to each other.

Multiplication works best with correlation. Where tested, correlation behaves best with Kronecker.

### Discriminativeness

The baseline `head` prefers different discriminativeness components depending on dimensionality. For models with $D < 5\,000$, SPMI is the best, while for other dimensions SCPMI is more competitive.

Addition and Kronecker benefit from SPMI, and multiplication from SCPMI, apart from the dimensionality of 10 000. H13, that PMI compression improves results in compositional tasks, is only supported by multiplication.

### 6.3.3 Difference between Max selection and heuristics on PhraseRel

Parameter selection based on heuristics is more stable than the one based on maximum values. However, in cases where different parameters are picked, there is little or no difference between these parameter choices. For example, studied similarity measures yield similar average performance for addition, see Figure 6.11a.

Manual heuristics do not pick the best result of 1 (Table A.5), but are close with a multiplicative model with 20 000 and 30 000 dimensions, yielding a score of 0.96 (Table A.6).

The average relative difference between the Max selection and the selection based on heuristics is 0.02 for `head`, 0.07 for addition, 0.04 for multiplication and 0.06 for Kronecker.

Over all compositional methods, the difference is 0.05, which is within the 10% limit set by H2.

## 6.4 Selected model transfer across the datasets

### 6.4.1 Difference between heuristics

There is little agreement on parameter selection based on heuristics among the three compositional datasets. The only consistent choice is global context probability (`cds`) and SCPMI discriminativeness for multiplicative models.

There is more pairwise agreement, for example, similarity based on correlation for additive models on KS14 and GS11 and $\log n$ frequency for multiplicative models between GS11 and PhraseRel. The pairwise agreement might be a sign of overfitting because there is no clear pattern. On the other side, the difference in performance between parameter choices might

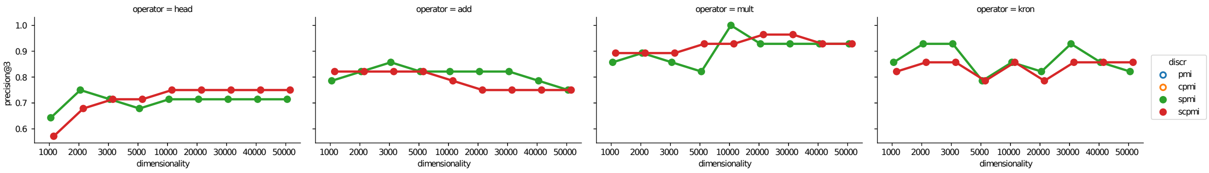


**Figure 6.12:** PhraseRel `discr`

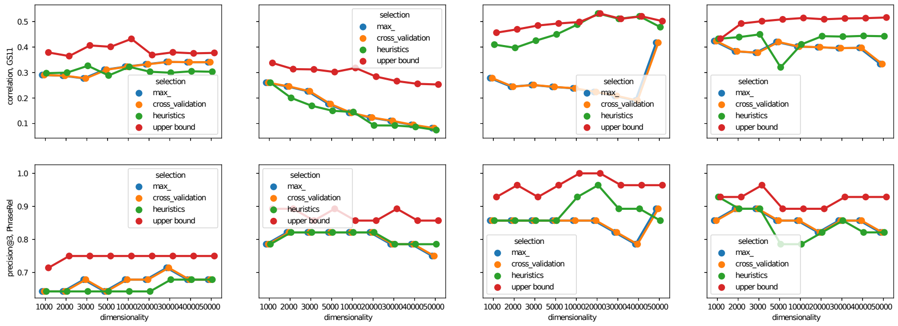


**Figure 6.13:** Transfer from KS14

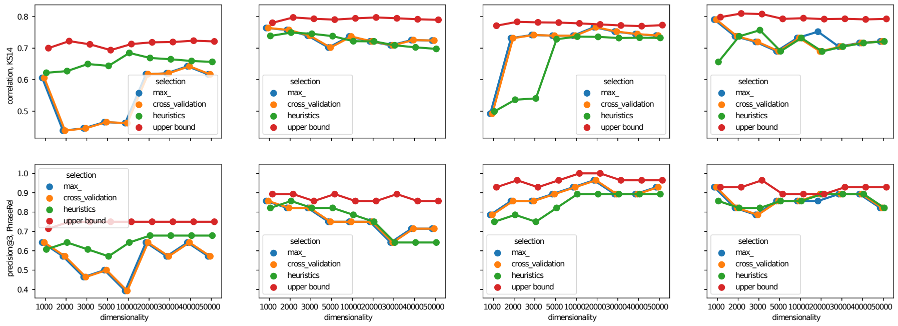


**Figure 6.14:** Transfer from GS11

be negligible, as some parameters consistently show low $R^2$ scores, for example `discr`. Consequently, there is inconsistency in the supported hypotheses.

### 6.4.2 Model transfer from KS14

Figure 6.13 shows the behaviour of models selected on the KS14 when they are transferred to GS11 and PhraseRel. During the transfer, there is little difference in performance between the selection methods, except in multiplicative models where heuristics show better performance and 5 000-dimensional Kronecker where heuristics give lower results than the Max-based selection.

Heuristic-based selection, on average, is closer to the upper bound than Max-based selection, supporting H1 that Max selection overfits. However, both are beyond the 10% boundary set by H2. When transferred to GS11, the average difference with the upper bound is 0.34 for Max and 0.24 for heuristics. When transferred to PhraseRel the average difference is 0.09 for Max and heuristics.

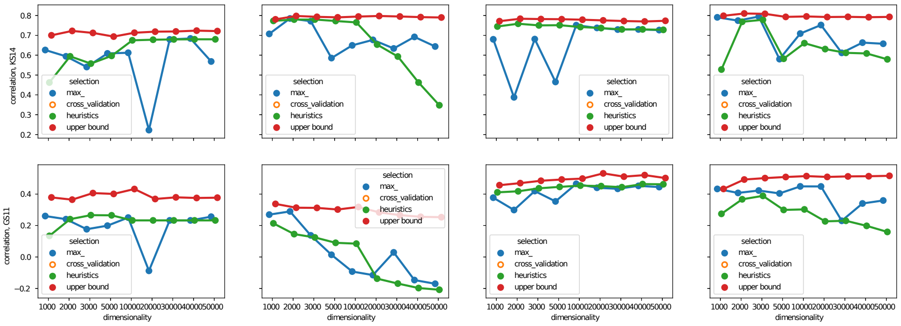


**Figure 6.15:** Transfer from Phraserel

## 6.4.3 Model transfer from GS11

Figure 6.14 shows that there is little difference between Max and heuristic-based selections. In the case of head composition, heuristics lead to higher performance, while for low-dimensional multiplicative models heuristics fall behind the Max selection on the KS14 dataset.

When GS11 models are transferred to KS14, the average difference with the upper bound is 0.12 and 0.11 for Max and heuristics respectively. For the transfer to PhraseRel, the differences are 0.13 for Max and 0.19 for heuristics. Again, the heuristic-based selection outperforms the Max based. This supports H1 that Max selection overfits, but the results are beyond the 10% limit of H2.

## 6.4.4 Model transfer from PhraseRel

Figure 6.15 shows that the performance of models based on PhraseRel is less stable, especially for selection by maximum performance.

Transfer to KS14 yields the average differences of 0.15 for Max and 0.14 for heuristics. Transfer to GS11 yields the average differences of 0.45 for Max and 0.51 for heuristics. Note that the transfer from PhraseRel to GS11 is the only case where Max selection, on average, is better than heuristics.

In general, over all compositional datasets, we see—in contrast to the lexical evaluation—that the Max-based selection might be prone to overfitting (H1). However, the result difference is far beyond the 10% limit set by H2, which might be due to the different nature of the tasks: similarity, disambiguation and relevance (compositional) versus similarity and relatedness (lexical).

## 6.5 Universal parameter selection for compositional datasets

To find parameters that are good for all three tasks, we combine their scores. As with lexical tasks, we normalise the scores on each dataset and report the average performance. The combined score is calculated the following way:

$$\text{score}_{\textit{compositional}}(m,o) = \frac{1}{3} \times \frac{\text{score}_{\textit{KS14}}(m,o)}{\max_m \text{score}_{\textit{KS14}}(m,o)} + \frac{1}{3} \times \frac{\text{score}_{\textit{GS11}}(m,o)}{\max_m \text{score}_{\textit{GS11}}(m,o)} + \frac{1}{3} \times \frac{\text{score}_{\textit{PhraseRel}}(m,o)}{\max_m \text{score}_{\textit{PhraseRel}}(m,o)} \tag{6.1}$$

where $m$ is a model and $o$ is an operator.

Figure A.1 shows the performance of the models based on the combined selection over the KS14, GS11 and PhraseRel datasets.

The performance of selected models together with the selected parameters is shown in Table A.7 (Max selection) and Table A.8 (selection based on heuristics).

The combined minimum significant difference for the three datasets is $\sigma^{0.9}_{0.05} = 0.14$.

### 6.5.1 Max selection

Models with many dimensions do not always perform better than their low-dimensional counterparts. Particularly, only head and multiplication benefit from the high number of dimensions. Addition and Kronecker are closer to the upper bound with dimensionality of a few thousand.

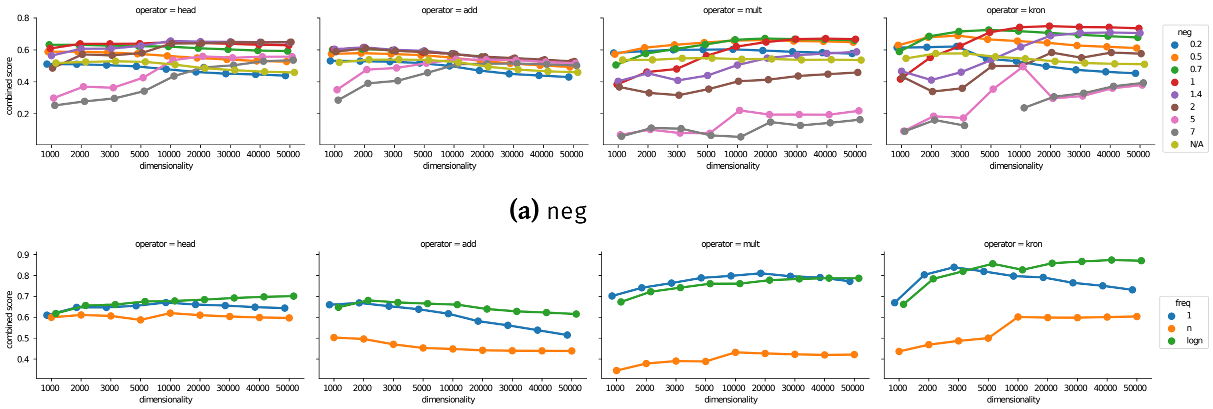
Regarding the hypotheses, there is support of H6 (non-constant frequency should be used with high-dimensional spaces) for addition, multiplication and Kronecker, H7 (context distribution smoothing is optimal for high-dimensional spaces) for multiplication and H8 (low-dimensional spaces should be dense, high-dimensional—sparse) for Kronecker.

### 6.5.2 Heuristics

The linear model achieves $R^2 = 0.77$. Table 6.4 shows the partial $R^2$ values for the parameters. The most influential parameters are neg, freq and compositional operator.

#### Neg

For the baseline operator head and models with $D < 10\,000$, the neg should be set to 1 (only 5 and 7 are statistically significantly different), otherwise, it should be 1.4, however 0.2 and



**(a)** `neg`

**(b)** `freq`

**Figure 6.16:** Compositional influence of `neg` and `freq`

no shifting is statistically significantly different for all dimensions and 7 for $D \geqslant 40\,000$ (Figure 6.16a).

For addition, 1 is the best choice of `neg`, but the performance of $k$ values follows H8: the more dimensions a model has, the sparser it should be. However, the only statistically significantly different results are with $k = 7$ with $D \leqslant 3\,000$, and $k = 5$ with $D = 1\,000$.

Multiplication benefits from denser spaces. If the dimensionality is less than 10 000, then `neg` should be set to 0.5 (however no shifting 0.2, and 0.7 are not statistically significantly different), otherwise 0.7 is a good choice (0.2, 0.5, 1, 1.4 and no shifting are not statistically significantly different), confirming H8.

| parameter | partial $R^2$ |
|---|---|
| neg | 0.40 |
| freq | 0.29 |
| operator | 0.21 |
| cds | 0.15 |
| similarity | 0.08 |
| discr | 0.06 |
| dimensionality | 0.05 |

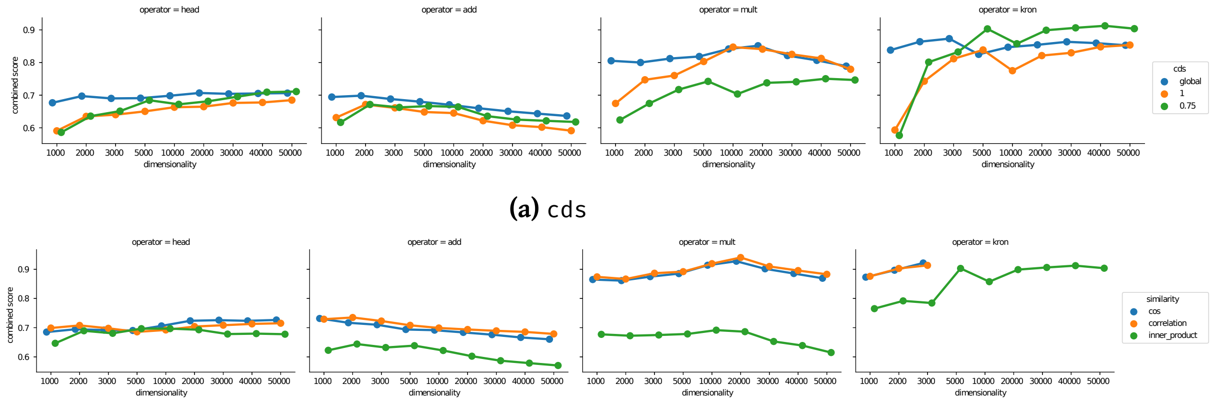
**Table 6.4:** Compositional feature ablation

Kronecker benefits from the `neg` of 0.7 if $D < 10\,000$ (0.5 and 1 are not statistically significantly different) and from 1 for the more dimensional cases (0.5, 0.7 and 1.4 are not statistically significantly different), also supporting H8. This is similar to multiplication, but Kronecker prefers less dense vectors.

## Freq

The frequency value of choice of all operators is $\log n$ with an exception of multiplication, where the constant frequency is preferred (Figure 6.16b).

For the baseline operator `head`, there is no statistically significant difference between parameter choices. For addition, multiplication and Kronecker, $n$ is statistically significantly worse



**(a)** `cds`

**(b)** Similarity

**Figure 6.17:** Compositional influence of `cds` and similarity

than the other choices.

For low-dimensional vector spaces ($D \leqslant 5\,000$), 1 behaves well, giving support of H6 that non-constant frequency is needed only with high-dimensional spaces.

### Context distribution smoothing

As Figure 6.17a shows, global context probability is the preferred choice of context probability in all cases, with an exception of Kronecker with $D > 3\,000$, where smoothed, local probabilities are better ($\alpha = 0.75$), supporting H7 that context distribution smoothing should be used with high-dimensional spaces. However, there is no statistically significant difference between the parameter choices.

### Similarity

Correlation is the dominant choice of the similarity measure (Figure 6.17b). However, cosine is preferred in the case of `head`, with $D > 5\,000$, and inner product is the only choice for composition with Kronecker with $D > 3\,000$.

There is no statistically significant difference between cosine and correlation for all compositional operators, which neither supports nor disputes H9 that expects correlation to outperform cosine with high-dimensional spaces.

The only statistically significant difference is between the inner product and the best score with multiplication.

### Discr

SPMI is the choice of `discr` that leads to the best average performance in most cases (Figure 6.18). However, the difference between SPMI and SCPMI is very small and there is no statistically significant difference between the parameter choices.

The exceptions are Multiplicative composition with $D \geqslant 10\,000$ and Kronecker with $D \leqslant 5\,000$ where SCPMI outperforms SPMI, as expected by H13 that PMI values compression is needed for composition.

### 6.5.3 Comparison with the selection based on one dataset

Manual selection based on a combination of the compositional datasets is more stable with regards to the chosen parameter values than the selection based on the highest values, even though manual selection does not always achieve the performance of Max selection, see Figure A.1.

The average difference with the upper bound is 0.04 for Max and heuristics, when applied to KS14. For GS11, the difference is 0.05 (Max) and 0.13 (Heuristics). For PhraseRel, the difference is 0.06 (Max) and 0.08 (Heuristics).

The numbers are much lower than the transfer of spaces selected on the basis of one dataset (Section 6.4). The average normalised difference is within the 10% limit (H2), with an exception of heuristics on GS11. This is evidence that there might be one universal model that fits various tasks (H4). The fact that the average normalised difference is smaller for Max-based selection is against H1 that expects Max selection to overfit. A combination of datasets that covers several phenomena (in our case, similarity, disambiguation and relatedness) might be more effective than manual heuristic-based selection.

The model selection procedures improve from the combination of datasets. One needs to keep in mind that in this case, we test model performance on the same dataset as we do parameter selection.

## 6.6 Conclusion

Phrasal experiments support most of the hypotheses stated in Section 3.3.

We see the confirmation of hypotheses on optimal parameter dependence on dimensionality (H5). In particular: H6 (non-constant frequency is beneficial with high-dimensional spaces) and H8 (high-dimensional spaces should be sparser). They are supported by all datasets and

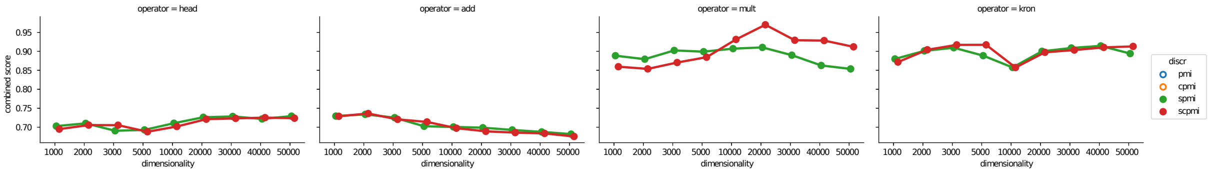


Figure 6.18: Compositional influence of `discr`

confirm H5. It is worth noting that even though an optimal choice of context distribution smoothing does not depend on dimensionality on KS14 and GS11, in the combined case the dependence holds.

Models that are selected on the experiments on a single dataset are prone to overfitting. Neither manual selection of parameters prevents it and the average normalised difference is above 10% on model transfer. This confirms H1 that Max selection overfits and rejects H2 that the relative gap in performance of the best model and a manually selected model is within 10%.

For the tested datasets, the observed differences are often not statistically significant. The minimum significant difference depends on the dataset size: the larger the dataset, the smaller the minimum difference. For KS14 with 108 items, the minimum significant difference is $\sigma^{0.9}_{0.05} = 0.07$. For GS11 with 200 items, the minimum significant difference is $\sigma^{0.9}_{0.05} = 0.05$. For PhraseRel—despite the fact that it consists of 644 query-document pairs—the minimum significant difference is huge $\sigma_{0.05} = 0.43$, because of a different evaluation metric. Such a high value makes the results statistically indistinguishable. Ideally, new datasets should be developed to evaluate compositional semantics models that make sure that the dataset size and the evaluation metric lead to a small minimum required difference.

Model selection based on the combination of datasets performs much better on each dataset (contrary to a single-dataset selected models). Both selection methods are within 10%, supporting H2.

Max selection models outperform heuristic selection models, suggesting that there is no overfitting in this case and H1 is not valid. In this case, the dataset combination covered three phenomena (similarity, disambiguation and relevance) and the precaution of overfitting by heuristics might be redundant.

This also suggests that there is a unique parameter choice that is universally applicable to compositional tasks (H4). The universal spaces are studied in Chapter 7.

# Chapter 7

# Universal models for both lexical and compositional tasks

Previously, we identified good models for specific datasets or task types: lexical and compositional. We managed to identify models that are good for either lexical tasks or compositional. This chapter investigates how well one model can perform on all tasks in contrast to the task-tailored models of previous chapters.

This is achieved by performing the evaluation on a combined score over the previous two lexical and three phrasal datasets. We not only combine the datasets together but also look for parameters that are good across all datasets with all compositional operators. Once optimal parameters are identified, they are tested on categorical compositional methods.

Even though the identified parameters have to compromise over different tasks (lexical and compositional) and compositional methods we achieve the new state-of-the-art results with Kronecker on KS14 and GS11. Moreover, the selected lexical representations improve over the results of the categorical compositional methods reported in the literature.

## 7.1 Operator-dependent universal models

First, we compute combined performance scores for the following operators: head, addition, multiplication and Kronecker. We normalise the scores of every dataset and weight lexical and compositional datasets equally. Within a dataset category, the datasets are weighted equally. Such a weighting scheme is still simple, it treats the task types equally and does not focus on a particular dataset.

The combined score for a model and an operator is computed as:

$$
\begin{aligned}
\text{score}_{\mathit{universal}}(m,o) = & \frac{1}{2} \times \left( \frac{1}{2} \times \frac{\text{score}_{\mathit{SimLex-999}}(m)}{\max_m \text{score}_{\mathit{SimLex-999}}(m)} + \frac{1}{2} \times \frac{\text{score}_{\mathit{MEN}}(m)}{\max_m \text{score}_{\mathit{MEN}}(m)} \right) + \\
& \frac{1}{2} \times \left( \frac{1}{3} \times \frac{\text{score}_{\mathit{KS14}}(m,o)}{\max_m \text{score}_{\mathit{KS14}}(m,o)} + \frac{1}{3} \times \frac{\text{score}_{\mathit{GS11}}(m,o)}{\max_m \text{score}_{\mathit{GS11}}(m,o)} + \frac{1}{3} \times \frac{\text{score}_{\mathit{PhraseRel}}(m,o)}{\max_m \text{score}_{\mathit{PhraseRel}}(m,o)} \right)
\end{aligned}
\tag{7.1}
$$

where $m$ stands for a model and $o$ is a compositional operator.

The combined minimum significant difference is $\sigma^{0.9}_{0.05} = 0.12$.

### 7.1.1 Max selection

Table A.9 shows the performance of the models evaluated with a combined score of the lexical and compositional datasets. Parameter selection is much more stable than on all previous Max-based selections. The dominant `freq` choice is $\log n$, cosine is the measure of choice for multiplication and Kronecker (if available). Correlation is the best similarity measure for the additive composition. The optimal choice of a similarity measure does not depend on dimensionality—this observation does not support H9 that expects such dependence.

Interestingly, when shifting is applied, dense spaces perform better: 1 and 0.7 are the optimal `neg` values. Multiplication supports H8 that high-dimensional spaces optimally perform with sparse models.

The preference of a compositional operator depends on a model. Addition with many dimensions gives the best results on lexical tasks: 0.38 on SimLex-999 and 0.76 on MEN. Kronecker, on the other side, gives the highest values on compositional datasets supporting H12 (the word order is important for compositional operators): 0.80 on KS14, 0.51 on GS11 and 0.96 on PhraseRel. Multiplication, however, is a good compromise between the two. It gives the highest "combined" score of 0.95.

| parameter | partial $R^2$ |
|---|---|
| freq | 0.32 |
| neg | 0.29 |
| similarity | 0.22 |
| cds | 0.10 |
| discr | 0.09 |
| dimensionality | 0.07 |
| operator | 0.05 |

**Table 7.1:** Universal (operator-dependent) feature ablation

### 7.1.2 Heuristics

Performance of the models selected manually is shown in Table A.10. Again, there is a lot of consistency between parameters. The linear model achieves $R^2 = 0.83$. The most influencing parameters are `freq`, `neg` and a similarity measure. See Table 7.1 for more details.

Heuristics for addition choose models that score the highest on lexical tasks: 0.38 on SimLex-999 and 0.76 on MEN (Table A.10). Moreover, with more than 20 000 dimensions, there is no difference between the selection procedures (Max or heuristics) of the additive and Kronecker-based models.

Kronecker is strong in compositional tasks scoring 0.80 on KS14, 0.52 on GS11 and 0.93 on PhraseRel, which is in line with H12 that word order is important (Table A.10).

Multiplication and Kronecker support H8: the optimal shifting value $k$ depends on the dimensionality. Addition and Kronecker are consistent with H7: low-dimensional spaces benefit from global context probabilities, while high-dimensional spaces benefit from smoothed context probabilities with $\alpha = 0.75$.

Multiplication, again, is a compromise between the two: it gives the highest combined score of 0.94. The highest Kronecker's combined score is 0.91, while addition's highest score is only 0.84.

Regarding the hypotheses, we clearly see on Figure A.3a that there is no statistically significant difference between 1 and log $n$ frequencies for the low-dimensional spaces, but log $n$ is the best choice for the high-dimensional spaces, which is consistent with H6.

Shifting performs also in accordance with H8: for low-dimensional spaces $k = 0.7$ or even $k = 0.5$ leads to the highest result, while for high-dimensional spaces $k = 1$ or $k = 1.4$ are optimal.

We also see that there is no statistically significant difference between the cosine and correlation similarity measures giving a weak support of H9 that correlation is optimal for high-dimensional models.

Addition and Kronecker work best with global context probabilities on low-dimensional spaces, but benefit from local probabilities ($\alpha = 0.75$) for high-dimensional spaces, supporting H7: context distribution smoothing is beneficial with high-dimensional spaces. Multiplication, however, does not follow H7, as $\alpha = 0.75$ leads to weak performance for all dimensions.

There is no support of H13: for all operators, there is little difference between SPMI and SCPMI, so compression of PMI values might not, in general, boost the performance of compositional models.

### 7.1.3 Comparison of the selection methods

On lexical tasks, there is little difference between the model selection methods, especially for spaces with more than 30 000 dimensions, as Figures A.2a and A.2b show.

The average relative differences for Max selection are 0.03 (SimLex-999), 0.01 (MEN), 0.03 (KS14), 0.11 (GS11) and 0.06 (PhraseRel). For manual heuristics, the differences are 0.05 (SimLex-999), 0.01 (MEN), 0.03 (KS14), 0.11 (GS11) and 0.07 (PhraseRel). The numbers between different selection methods are close, with the exceptions of SimLex-999 (where Max selection is 0.01 points lower), and GS11 (where Max is lower by 0.05 points).

The high relative difference on GS11 is due to a poor performance of addition and `head`. The average normalised difference for addition is 0.22 for Max selection and for heuristics. For the baseline compositional operator `head`, the differences are 0.11 and 0.14 respectively. The differences for multiplication and Kronecker are less than 0.07, which is in accordance with the 10% margin of H2.

Contrary to H1, Max selection does not overfit, probably due to a broad selection of evaluation datasets.

When the models that are selected on lexical tasks are applied in a compositional setting, they perform worse than the models selected based on the universal score. This suggests that a model that is good for lexical tasks will not necessarily perform well on a compositional task, rejecting H11.

In addition, the difference between the good lexical models and the upper bound increases as dimensionality increases. This is the case for multiplication, the most notable difference is observed on KS14 (Figure A.2c).

It worth noting that on compositional tasks dimensionality does not contribute as much as on lexical tasks, with an exception of addition on the GS11 dataset, where performance decreases as dimensionality increases.

## 7.2 An operator-independent universal model

In the previous section, we have seen that even though parameter selection varies between operators, there are parameter choices that are shared, for example, correlation is the best similarity measure for addition, multiplication and Kronecker (if $D \leqslant 3\,000$). Given this and the fact that the difference between some of the choices is marginal, we try to look for a truly universal parameter combination. The aggregated score of a model is computed as:

$$\begin{aligned}
\text{score}_{\mathit{universal}}(m) = &\frac{1}{2}\left(\frac{1}{2}\frac{\text{score}_{\mathit{SimLex-999}}(m)}{\max_m \text{score}_{\mathit{SimLex-999}}(m)} + \frac{1}{2}\frac{\text{score}_{\mathit{MEN}}(m)}{\max_m \text{score}_{\mathit{MEN}}(m)}\right) + \\
&\frac{1}{2}\left(\frac{1}{6}\frac{\text{score}_{\mathit{KS14}}(m, \mathit{add})}{\max_m \text{score}_{\mathit{KS14}}(m, \mathit{add})} + \frac{1}{6}\frac{\text{score}_{\mathit{GS11}}(m, \mathit{add})}{\max_m \text{score}_{\mathit{GS11}}(m, \mathit{add})} + \frac{1}{6}\frac{\text{score}_{\mathit{PhraseRel}}(m, \mathit{add})}{\max_m \text{score}_{\mathit{PhraseRel}}(m, \mathit{add})} + \right. \\
&\left. \frac{1}{6}\frac{\text{score}_{\mathit{KS14}}(m, \mathit{mult})}{\max_m \text{score}_{\mathit{KS14}}(m, \mathit{mult})} + \frac{1}{6}\frac{\text{score}_{\mathit{GS11}}(m, \mathit{mult})}{\max_m \text{score}_{\mathit{GS11}}(m, \mathit{mult})} + \frac{1}{6}\frac{\text{score}_{\mathit{PhraseRel}}(m, \mathit{mult})}{\max_m \text{score}_{\mathit{PhraseRel}}(m, \mathit{mult})}\right)
\end{aligned} \tag{7.2}$$

where *add* stands for addition and *mult* stands for multiplication. We do not test Kronecker, because it is not tested on all dimensions with the cosine and correlation similarity measures. We also exclude the baseline operator `head`.

### 7.2.1 Max selection

Table A.11 shows the combined scores for all datasets abstracting over a compositional operator.

The parameter selection shows a clear pattern. Low-dimensional spaces perform best with 1 as the frequency choice, while high-dimensional models perform better with $\log n$, confirming to H6 that non-linear frequency should be used with high-dimensional models. Cosine is a better-suited similarity measure for models with few dimensions and correlation is suited to those with many, which is in line with H9. Finally, global context probabilities are the best in a low-dimensional case, while local context probabilities perform best with many dimensions, supporting H7.

### 7.2.2 Heuristics

The linear model gives $R^2 = 0.90$. The most influential parameters are `freq`, similarity measure and `neg`, refer to Table 7.2.

Heuristics, in general, repeat the parameter choices of Max selection, but the switch between parameter values happens at a higher number of dimensions (at 20 000, not at 5 000). Refer to Table A.12 for the results and Figure A.4 for the parameter behaviour.

| parameter | partial $R^2$ |
|---|---|
| freq | 0.43 |
| similarity | 0.27 |
| neg | 0.20 |
| discr | 0.09 |
| cds | 0.09 |
| dimensionality | 0.03 |

**Table 7.2:** Universal (operator-independent) feature ablation

The average normalised differences with the upper bound for Max selection are 0.05 (SimLex-999), 0.03 (MEN), 0.06 (KS14), 0.11 (GS11) and 0.12 (PhraseRel). The differences for heuristics are in general higher: 0.06 (SimLex-999), 0.05 (MEN), 0.08 (KS14), 0.09 (GS11) and 0.14 (PhraseRel).

Max selection is above the 10% margin of H2 on GS11 and PraseRel, while heuristics are above the margin only on PhraseRel.

## 7.3 Experiments with categorical compositional operators

Sections 7.1 and 7.2 identified models that perform well on a range of tasks. The majority of them are within the 10% margin set by H2. We apply selected models with tensor-based operators on phrasal datasets (KS14, GS11 and PhraseRel).

| Opertor | KS14 | GS11 | PhraseRel |
|---|---|---|---|
| Add | **0.79** | 0.34 | **0.89** |
| Mult | **0.77** | **0.51** | **1.00** |
| Kron | **0.80** | **0.52** | **0.96** |
| Relational | **0.77** | 0.39 | **0.89** |
| Copy-object | 0.63 | 0.28 | **0.82** |
| Copy-subject | **0.74** | 0.40 | **0.82** |
| Frobenius-add | **0.76** | 0.37 | **0.82** |
| Frobenius-mult | **0.75** | 0.30 | **0.86** |
| Frobenius-outer | **0.77** | 0.38 | **0.89** |

**Table 7.3:** The best scores on compositional tasks based on the universal selections. There is no statistically significant difference between the values in bold.

Table 7.3 shows the best results we obtained for each operator, Tables A.13, A.14 and A.15 show all the results together with the model parameters and Figure A.5 depicts the data.

In general, the best results are achieved with 3 000-dimensional models (with an exception on GS11 where 1 000-dimensional models perform better in 4 out of 6 cases, and copy-subject on KS14). Also, performance increases as dimensionality increases.

Max selection based on Kronecker leads to the highest results. The exceptions are Frobenius multiplication and copy-subject on KS14, where the model that is best with addition also leads to the highest results among tensor-based composition. On PhraseRel, copy-subject performs best with operator-independently selected space.

Relational is the fourth best compositional operator after addition, multiplication or Kronecker on KS14 (0.77) and PhraseRel (0.89). Copy-subject is the best on GS11 (0.40). Frobenius-outer gives the highest result on PhraseRel together with relational.

On GS11 and PhraseRel, newly tested operators outperform addition, whose scores are 0.34 and 0.89, respectively.

While there is a difference between selection methods, there are no clear outliers and the models show similar behaviour.

## 7.4 Putting results into perspective

This section discusses the results of the experiments in the context of our preliminary studies and the work of others.

In an earlier study (Milajevs et al. 2014), we compared a PPMI-weighted space, an LMI-weighted and SVD-reduced space and a space based on the original word2vec vectors obtained from the Google News corpus (Mikolov et al. 2013b). The same compositional operators were evaluated as in this thesis. The count-based models selected in the earlier study were the ones that were considered to be efficient in the compositional tasks and were used in the studies that introduced the evaluation datasets: KS14 (Kartsaklis and Sadrzadeh 2014) and GS11 (Grefenstette and Sadrzadeh 2011a). Thus, Milajevs et al. (2014) can be seen as a replication of the experiments in previous papers. The experiments showed that on small-scale tasks (KS14 and GS11) count-based models are competitive with neural word vectors, however, word2vec vectors are superior in dialog act tagging and paraphrase detection.

In that study, additive composition with an SVD-reduced space gave the best result of 0.73 on KS14. The best tensor-based result was 0.66, achieved with word2vec and copy-object. All our models (Table 7.3), with the exception of copy-object, statistically significantly improve over our previous best scores. Despite being lower, our copy-object (0.63) is not statistically significantly different from the word2vec score reported in Milajevs et al. (2014).

On the GS11 dataset, systematic parameter selection leads to spaces that improve over the corresponding operators in the earlier study on all but two of the compositional methods (the exceptions being copy-object and Frobenius mult). In addition to that, multiplication and Kronecker statistically significantly improve over the overall best-reported score of 0.46 in Milajevs et al. (2014). Kronecker yields the highest score of 0.52.

Kim et al. (2015b) adopt the evaluation procedure of Milajevs et al. (2014) to test an extended word2vec model that is tuned for multiplicative interaction of the vectors, not additive, as the original word2vec. They improve on most of the composition operators on the KS14 and GS11 datasets.

They achieve the best result of 0.77 with addition on KS14. Three of our models (addition, multiplication and Kronecker) outperform that score, however the difference is not statistically significant. Also, our results are better than the results reported by comparing results by operator (our result are shown in brackets for comparison): multiplication 0.44 (0.77, statistically significant), Kronecker 0.62 (0.80, statistically significant), relational 0.67 (0.77, statistically significant), copy-subject 0.45 (0.74, statistically significant), copy-object 0.61 (0.63, not statistically significant), Frobenius addition 0.61 (0.76, statistically significant), Frobenius multiplication 0.61 (0.75, statistically significant), Frobenius outer 0.66 (0.77, statistically significant).

On G11, their best score is 0.39, which is lower than the results that we get with multiplication (0.51, statistically significant), Kronecker (0.52, statistically significant), relational (0.40, not statistically significant) and copy-subject (0.40, not statistically significant). However, we get lower results with copy-object (0.29, statistically significant) and Frobenius outer (0.39, not statistically significant).

Hashimoto and Tsuruoka (2015) learn the matrices of transitive verbs using implicit tensor factorisation. The verb matrices are learned in two ways: one only takes into account the verb arguments (the subject and the object, referred as SVO in the paper), another, in addition to that, employs the adjuncts that complement the meaning of the verb phrases (SVOPN). They use copy-subject as the compositional operator. The main baseline is their previous method described in Hashimoto et al. (2014).

In comparison to their SVO results, our are higher: they get 0.48 on GS11 (no statistically significantly difference) and 0.48 on KS14 (statistically significantly different). These results are lower than our Multiplication and Kronecker on GS11 and all operators on KS14.

The SVOPN model with the score of 0.61 statistically significantly outperforms our best (0.52) on GS11. While they get a higher score on KS14 of 0.74 (though it is obtained with the Hashimoto et al. (2014) baseline) it is still lower than all of our results, except copy-subject and copy-object, but the improvement is not statistically significant. Interestingly, our result of 0.74 with copy-subject is close to their score but is still lower.

Hashimoto and Tsuruoka (2016) jointly learn compositional and non-compositional phrase embeddings by using a compositionality scoring function. They improve on their previous work and get the score of 0.68 on GS11 versus our 0.52 with a statistically significant difference.

Fried et al. (2015) use low-rank tensors to approximate third-order tensors of verbs. They achieve the scores of 0.47 on GS11 and 0.68 on KS14 with categorical composition and 0.71 on KS14 with addition. While our best result is statistically significantly better on GS11, categorical operators score lower, the best is copy-subject (0.40). On KS14, our experiments produce higher results (with an exception of copy-object) than their best (additive) model, addition and Kronecker make a statistically significant improvement. It is worth noting the study of Polajnar et al. (2015) uses discourse features to build vectors, but the experiment results reported there are not compatible with ours because we averaged the human-provided scores before computing correlation, while they treat each human score individually without averaging.

Overall, we improved over the scores by identifying a better set of model parameters rather than developing a more sophisticated model.

## 7.5 Conclusion

This chapter identified a few spaces that work well on a broad range of lexical and compositional tasks.

Despite the expectation in H1, we see that Max selection does not overfit if the models are evaluated on diverse tasks. In fact, heuristics become too conservative in this case, however, we still suggest manual analysis when a small number of datasets is used.

Our universal models perform within the 10% margin of H2 in the majority of the experiments. Moreover, the operator-independent universal space is competitive with spaces that were selected with an operator in mind, supporting the idea that there is a universal vector space for all kind of tasks and H4.

The selections show that an optimal parameter choice depends on dimensionality (H5). As we have seen in Section 7.1.3, a good lexical model might fail in a compositional setting (H11).

We have also seen in Section 7.1.1 that good lexical models favour the additive composition, while Kronecker is more optimal for composition and multiplication is a compromise between the two. This, and the fact that a similarity has been found between lexical and compositional tasks (Kiela and Clark 2014, Section 4), might be an explanation of why addition is considered to be the best compositional operator. In our experiments, multiplication and Kronecker consistently outperform addition.

While parameter selection depends on dimensionality, the performance on compositional task depends on it to a much lower extent.

Our selection methodology has produced some statistically significantly better results in the KS14 dataset over widely used count-based vectors (Milajevs et al. 2014), neural vectors in a compositional setting (Kim et al. 2015b, Milajevs et al. 2014) and learned verb tensors (Fried et al. 2015, Hashimoto and Tsuruoka 2015; 2016).

While our results on GS11 are close to the current state-of-the-art results (Hashimoto and Tsuruoka 2016), there is room for improvement, especially for tensor-based compositional operators. The difference in the performance might be explained as the limit of the count-based methods or the unexplored, and therefore untuned, parameters of the verb matrix. For example, we consider all different subject-object occurrences despite their frequency in the corpus. Using only the subject-object pairs that appeared at least 100 times might improve the results.

The gap between the multiplicative and Kronecker composition in our work indicates that the categorical methods can be improved. We see that the word order is important in the task, otherwise, Kronecker would not outperform addition and multiplication. Because categorical methods take word order into account, there is a potential for them to improve. However, it is not clear whether the verb matrices are of good quality. The verb matrices obtained by different ways need to be tested on a lexical similarity task, for example Gerz et al. (2016). Also, the similarity judgments of verbs in SimLex-999 and MEN can be used.

# Chapter 8

# Conclusion

This work is a systematic study of vector space models for similarity estimation, based on the distributional hypothesis (Harris 1954) and Frege's principle of compositionality (Coecke et al. 2010, Janssen 2001). The goal of this work is to provide performance numbers of distributional models that are robust to overfitting and are representative of this kind of method.

Another goal of the current study is to compare the parameters within the distributional approach and identify parameter combinations that lead to high performance of the corresponding models.

The experiments in the study are performed on two lexical tasks—SimLex-999 (Hill et al. 2015) and MEN (Bruni et al. 2014)—and three phrasal tasks—KS14 (Kartsaklis and Sadrzadeh 2014), GS11 (Grefenstette and Sadrzadeh 2011b) and PhraseRel (Section 5). In addition to individual dataset evaluation, the models' performance scores are combined to identify models that perform well on the collections of tasks, namely lexical (Section 4.4), compositional (Section 6.5) and universal (Chapter 7).

The vector component values are based on the PMI quantification of the co-occurrence counts. To minimise the effect of noise in the co-occurrence data, the PMI score itself is modified by weighting, shifting, compression and others—see Section 3.2.1 for more details. We identify an optimal parameter choice based on dimensionality of the underlying vector space. In compositional tasks, we experiment with point-wise operators (addition and multiplication) and with categorical operators (Section 7.3, Coecke et al. (2010)).

**Representative performance of count-based distributional methods** As a result of a systematic study, we identified parameters of distributional models that replicate results obtained in lexical experiments (Table 4.10) and achieve the new state-of-the-art result on the sentence similarity task (KS14, Table 7.3) with Kronecker-based composition. In addition to that, the performance of categorical compositional methods was improved.

The model parameters we have identified are competitive with other meaning models, for example, predictive models (Mikolov et al. 2013a).

**Optimal parameter choice** The experiments (Chapters 4, 6 and 7) show that the optimal parameter choice depends on dimensionality. While there are more optimal choices for particular datasets, we suggest using SCPMI with $\log n$ frequency, global context probabilities, shifted $k = 0.7$ values and correlation as the similarity measure with at least 20 000 dimensional space. For compositional tasks, we suggest 3 000 dimensions and cosine as the similarity measure, keeping other parameters the same.

High-dimensional models contain more noise signal because by design they include co-occurrence counts of lower frequencies. The $\log n$ frequency component, context distribution smoothing and PMI value shifting minimise the influence of noise that presents in the co-occurrence data. In contrast to high-dimensional models, low-dimensional models are based on less noisy data, making noise handling unnecessary.

The fact that we were able to identify parameters that perform on a variety of tasks suggests that there might be a single model that is good in a variety of tasks (Pereira et al. 2016).

**Lexical representations in compositional setting** In Section 7.1 we observed a direct link between model performance and the combination of lexical parameters with a compositional operator. Kronecker-optimised lexical parameters perform best on compositional tasks and underperform on lexical tasks. Addition-based lexical parameters perform best on lexical tasks and match parameters that are based on word similarity tasks. Multiplication-based optimal parameters is a good compromise as they achieve a balance in performance on lexical and compositional tasks.

The fact that addition-based optional parameters follow the parameters that are the best for word similarity estimation biases against other compositional operators especially when iterative parameter tuning is used.

**Parameter selection procedure** Two model selection procedures were tested to avoid overfitting. One selects the parameters that lead to the highest performance, while the other performs selection based on the average performance of the parameter values. We see that if a single dataset is used for model selection, that the best model is overfit and suggest using a more elaborated parameter selection technique. However, when the selection is based on a combination of datasets, then the Max-based selection picks models that are less likely to overfit.

**Future work** There are several directions for future work. First of all, we explored unreduced spaces, for example, we did not apply SVD (Bullinaria and Levy 2012). Apart from a particular dimensionality reduction method being superior, it might interact with other parameters (Lapesa and Evert 2014) and change the optimal values, in contrast to the reasoning of Kiela and Clark (2014) that "...dimensionality reduction relies on some original non-reduced model, and directly depends on its quality."

Another direction is the experimentation on a a larger number of datasets. While more datasets are being proposed, for example Gerz et al. (2016), and the current datasets are being criticised,[1] it is important to have datasets that share the same goal (for example, provide similarity judgements), but are constructed by different groups and employ different methods during the dataset construction.

While categorical compositional methods are built on solid theoretical grounds (Coecke et al. 2010) and have been shown previously (Fried et al. 2015, Grefenstette and Sadrzadeh 2011b, Hashimoto and Tsuruoka 2016, Kartsaklis and Sadrzadeh 2014, Kim et al. 2015b) and in this work to be competitive with other methods, more parameter exploration work has to be done, especially for the way how the verb (or other relational) tensors are built. On the theoretical side, the categorical methods need to be of lower computation complexity, as the current way of building verb matrices is not feasible for vectors over 3 000 components.

Significance testing should become a routine in the field of distributional semantics. The current datasets are not designed with such a requirement in mind, making many of the reported improvements in the literature and in this thesis statistically insignificant. To overcome this, new datasets should be designed that make sure that the evaluation procedure and the dataset size lead to a small minimum statistical difference (Faruqui et al. 2016, Rastogi et al. 2015).

[1] Refer to The 1st Workshop on Evaluating Vector-Space Representations for NLP at `https://www.aclweb.org/portal/content/1st-workshop-evaluating-vector-space-representations-nlp`.

# Bibliography

Jacob Andreas and Dan Klein. How much do word embeddings encode about syntax? In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 822–827, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P14-2133.

Marco Baroni and Alessandro Lenci. How we blessed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10, Edinburgh, UK, July 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W11-2501.

Marco Baroni and Roberto Zamparelli. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1183–1193, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1870658.1870773.

Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 23–32, Avignon, France, April 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/E12-1004.

Marco Baroni, Raffaela Bernardi, and Roberto Zamparelli. Frege in space: A program of compositional distributional semantics. *LiLT (Linguistic Issues in Language Technology)*, 9, 2014a. URL http://csli-lilt.stanford.edu/ojs/index.php/LiLT/article/view/6.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland, June 2014b. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P14-1023.

Miroslav Batchkarov, Thomas Kober, Jeremy Reffin, Julie Weeds, and David Weir. A critique of word similarity as a method for evaluating distributional semantic models. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 7–12, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://anthology.aclweb.org/W16-2502.

Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer, 2006. URL http://machinelearning.wustl.edu/mlpapers/paper_files/BengioDVJ03.pdf.

William Blacoe and Mirella Lapata. A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 546–556, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=2390948.2391011.

Johan Bos. Wide-Coverage Semantic Analysis with Boxer. In Johan Bos and Rodolfo Delmonte, editors, *Semantics in Text Processing. STEP 2008 Conference Proceedings*, volume 1 of *Research in Computational Semantics*, pages 277–286. College Publications, 2008. URL http://www.aclweb.org/anthology/W08-2222.

Johan Bos and Malte Gabsdil. First-order inference and the interpretation of questions and answers. *Proceedings of Götalog*, pages 43–50, 2000.

N. Bourbaki. *Commutative Algebra: Chapters 1-7*, volume 1. Springer Science & Business Media, 1998.

Elia Bruni, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 136–145, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=2390524.2390544.

Elia Bruni, Nam Khanh Tran, and Marco Baroni. Multimodal distributional semantics. *J. Artif. Int. Res.*, 49(1):1–47, January 2014. ISSN 1076-9757. URL http://dl.acm.org/citation.cfm?id=2655713.2655714.

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, 2007. ISSN 1554-3528. doi: 10.3758/BF03193020. URL http://dx.doi.org/10.3758/BF03193020.

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. doi: 10.3758/s13428-011-0183-8. URL http://dx.doi.org/10.3758/s13428-011-0183-8.

Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. In *Proceedings of the 27th Annual Meeting on Association for Computational Linguistics*, ACL '89, pages 76–83, Stroudsburg, PA, USA, 1989. Association for Computational Linguistics. doi: 10.3115/981623.981633. URL https://doi.org/10.3115/981623.981633.

Kenneth Ward Church and Patrick Hanks. Word association norms mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, 1990. URL http://www.aclweb.org/anthology/J90-1003.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. Mathematical foundations for a compositional distributional model of meaning. *CoRR*, abs/1003.4394, 2010. URL http://arxiv.org/abs/1003.4394.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390177. URL http://doi.acm.org/10.1145/1390156.1390177.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. Minimal recursion semantics: An introduction. *Research on Language and Computation*, 3(2):281–332, 2005. ISSN 1572-8706. doi: 10.1007/s11168-006-6327-9. URL http://dx.doi.org/10.1007/s11168-006-6327-9.

Tom Dietterich. Overfitting and undercomputing in machine learning. *ACM Comput. Surv.*, 27 (3):326–327, September 1995. ISSN 0360-0300. doi: 10.1145/212094.212114. URL http://doi.acm.org/10.1145/212094.212114.

Georgiana Dinu and Mirella Lapata. Measuring distributional similarity in context. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1162–1172, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1870658.1870771.

William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*. Asia Federation of Natural Language Processing, 2005. URL http://research.microsoft.com/apps/pubs/default.aspx?id=101076.

David R. Dowty, Robert E. Wall, and Stanley Peters. *Introduction to Montague Semantics*. Springer Netherlands, 1980. doi: 10.1007/978-94-009-9065-4. URL http://dx.doi.org/10.1007/978-94-009-9065-4.

Stefan Evert. *The statistics of word cooccurrences: word pairs and collocations*. PhD thesis, Universität Stuttgart, Holzgartenstr. 16, 70174 Stuttgart, 2005. URL http://elib.uni-stuttgart.de/opus/volltexte/2005/2371.

Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. Problems with evaluation of word embeddings using word similarity tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 30–35, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://anthology.aclweb.org/W16-2506.

Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. Introducing and evaluating ukWaC, a very large web-derived corpus of English. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google*, pages 47–54, 2008.

Samuel Fillenbaum and Amnon Rapoport. Verbs of judging, judged: A case study. *Journal of Verbal Learning and Verbal Behavior*, 13(1):54 – 62, 1974. ISSN 0022-5371. doi: 10.1016/S0022-5371(74)80030-7. URL http://www.sciencedirect.com/science/article/pii/S0022537174800307.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: The concept revisited. *ACM Trans. Inf. Syst.*, 20(1):116–131, January 2002. ISSN 1046-8188. doi: 10.1145/503104.503110. URL http://doi.acm.org/10.1145/503104.503110.

John R. Firth. A Synopsis of Linguistic Theory, 1930-1955. *Studies in Linguistic Analysis*, pages 1–32, 1957.

Daniel Fried, Tamara Polajnar, and Stephen Clark. Low-rank tensors for verbs in compositional distributional semantics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 731–736, Beijing, China, July 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P15-2120.

D. Gerz, I. Vulić, F. Hill, R. Reichart, and A. Korhonen. SimVerb-3500: A Large-Scale Evaluation Set of Verb Similarity. *ArXiv e-prints*, August 2016. URL http://arxiv.org/abs/1608.00869.

Nelson Goodman. Problems and projects. 1972.

E. Grefenstette, G. Dinu, Y. Zhang, M. Sadrzadeh, and M. Baroni. Multi-step regression learning for compositional distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 131–142, Potsdam, Germany, March 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W13-0112.

Edward Grefenstette and Mehrnoosh Sadrzadeh. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1394–1404, Stroudsburg, PA, USA, 2011a. Association for Computational Linguistics. ISBN 978-1-937284-11-4. URL http://dl.acm.org/citation.cfm?id=2145432.2145580.

Edward Grefenstette and Mehrnoosh Sadrzadeh. Experimenting with transitive verbs in a DisCoCat. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, GEMS '11, pages 62–66, Stroudsburg, PA, USA, 2011b. Association for Computational Linguistics. ISBN 978-1-937284-16-9. URL http://dl.acm.org/citation.cfm?id=2140490.2140497.

Ulrike Hahn. Similarity. *Wiley Interdisciplinary Reviews: Cognitive Science*, 5(3):271–280, 2014. ISSN 1939-5086. doi: 10.1002/wcs.1282. URL http://dx.doi.org/10.1002/wcs.1282.

Ulrike Hahn and Nick Chater. Concepts and similarity. *Knowledge, concepts and categories*, pages 43–92, 1997. URL https://books.google.co.uk/books?hl=en&lr=&id=pc3XAQAAQBAJ&oi=fnd&pg=PA43&dq=Concepts+and+similarity.&ots=it4aC91-qv&sig=ebsOGf72FEIpE0WX6HZoWhi2SsQ#v=onepage&q=Concepts%20and%20similarity.&f=false.

Guy Halawi, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren. Large-scale learning of word relatedness with constraints. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 1406–1414, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1462-6. doi: 10.1145/2339530.2339751. URL http://doi.acm.org/10.1145/2339530.2339751.

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

Kazuma Hashimoto and Yoshimasa Tsuruoka. Learning embeddings for transitive verb disambiguation by implicit tensor factorization. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 1–11, Beijing, China, July 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W15-4001.

Kazuma Hashimoto and Yoshimasa Tsuruoka. Adaptive joint learning of compositional and non-compositional phrase embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 205–215, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P16-1020.

Kazuma Hashimoto, Pontus Stenetorp, Makoto Miwa, and Yoshimasa Tsuruoka. Jointly learning word representations and composition functions using predicate-argument structures. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1544–1555, Doha, Qatar, October 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D14-1163.

Nancy M. Henley. A psychological study of the semantics of animal terms. *Journal of Verbal Learning and Verbal Behavior*, 8(2):176 – 184, 1969. ISSN 0022-5371. doi: 10.1016/S0022-5371(69)80058-7. URL http://www.sciencedirect.com/science/article/pii/S0022537169800587.

Karl Moritz Hermann and Phil Blunsom. The role of syntax in vector space models of compositional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 894–904, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P13-1088.

Felix Hill, Roi Reichart, and Anna Korhonen. Simlex-999: Evaluating semantic models with genuine similarity estimation. *Comput. Linguist.*, 41(4):665–695, December 2015. ISSN 0891-2017. doi: 10.1162/COLI_a_00237. URL http://dx.doi.org/10.1162/COLI_a_00237.

Theo M. V. Janssen. Montague semantics. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Spring 2016 edition, 2016. URL http://plato.stanford.edu/archives/spr2016/entries/montague-semantics/.

Theo M.V. Janssen. Frege, contextuality and compositionality. *Journal of Logic, Language and Information*, 10(1):115–136, 2001. ISSN 1572-9583. doi: 10.1023/A:1026542332224. URL http://dx.doi.org/10.1023/A:1026542332224.

Nal Kalchbrenner and Phil Blunsom. Recurrent convolutional neural networks for discourse compositionality. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 119–126, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W13-3214.

Dimitri Kartsaklis and Mehrnoosh Sadrzadeh. Prior disambiguation of word tensors for constructing sentence vectors. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1590–1601, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D13-1166.

Dimitri Kartsaklis and Mehrnoosh Sadrzadeh. A study of entanglement in a categorical framework of natural language. In *Proceedings of the 11th Workshop on Quantum Physics and Logic (QPL)*, Kyoto, Japan, June 2014.

Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, and Stephen Pulman. A unified sentence space for categorical distributional-compositional semantics: Theory and experiments. In *Proceedings of COLING 2012: Posters*, pages 549–558, Mumbai, India, December 2012. The COLING 2012 Organizing Committee. URL http://www.aclweb.org/anthology/C12-2054.

Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, and Stephen Pulman. Separating disambiguation from composition in distributional semantics. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 114–123, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W13-3513.

Douwe Kiela and Stephen Clark. A systematic study of semantic vector space model parameters. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 21–30, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W14-1503.

Joo-Kyung Kim, Marie-Catherine de Marneffe, and Eric Fosler-Lussier. Neural word embeddings with multiplicative feature interactions for tensor-based compositions. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 143–150, Denver, Colorado, June 2015a. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W15-1520.

Joo-Kyung Kim, Marie-Catherine de Marneffe, and Eric Fosler-Lussier. Neural word embeddings with multiplicative feature interactions for tensor-based compositions. In *Proceedings of NAACL-HLT*, pages 143–150, 2015b. URL http://www.aclweb.org/anthology/W15-1520.

S. Kim, W. J. Wilbur, and Z. Lu. Bridging the Gap: a Semantic Similarity Measure between Queries and Documents. *ArXiv e-prints*, August 2016. URL http://arxiv.org/abs/1608.01972.

Walter Kintsch. Predication. *Cognitive Science*, 25(2):173 – 202, 2001. ISSN 0364-0213. doi: http://dx.doi.org/10.1016/S0364-0213(01)00034-9. URL http://www.sciencedirect.com/science/article/pii/S0364021301000349.

Gabriella Lapesa and Stefan Evert. Evaluating neighbor rank and distance measures as predictors of semantic priming. In *Proceedings of the Fourth Annual Workshop on Cognitive Modeling and Computational Linguistics (CMCL)*, pages 66–74, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W13-2608.

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. URL http://www.aclweb.org/anthology/Q14-1041.

Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2177–2185. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5477-neural-word-embedding-as-implicit-matrix-factorization.pdf.

Omer Levy, Yoav Goldberg, and Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015. ISSN 2307-387X. URL https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/570.

David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. RCV1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397, 2004.

Bill MacCartney and Christopher D. Manning. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, RTE '07, pages 193–200, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1654536.1654575.

Christopher Manning and Hinrich Schuetze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999. URL https://mitpress.mit.edu/books/foundations-statistical-natural-language-processing.

Arthur B. Markman and Dedre Gentner. Commonalities and differences in similarity comparisons. *Memory & Cognition*, 24(2):235–249, 1996. ISSN 1532-5946. doi: 10.3758/BF03200884. URL http://dx.doi.org/10.3758/BF03200884.

Douglas L Medin, Robert L Goldstone, and Dedre Gentner. Respects for similarity. *Psychological review*, 100(2):254, 1993. doi: 10.1037/0033-295X.100.2.254. URL http://psycnet.apa.org/journals/rev/100/2/254/.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a. URL http://arxiv.org/pdf/1301.3781.pdf.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013b. URL http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL-HLT*, pages 746–751, 2013c. URL http://www.aclweb.org/anthology/N13-1090.pdf.

Dmitrijs Milajevs and Sascha Griffiths. A proposal for linguistic similarity datasets based on commonality lists. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 127–133, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-2523. URL http://anthology.aclweb.org/W16-2523.

Dmitrijs Milajevs and Matthew Purver. Investigating the contribution of distributional semantic information for dialogue act classification. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 40–47, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-1505. URL http://www.aclweb.org/anthology/W14-1505.

Dmitrijs Milajevs, Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, and Matthew Purver. Evaluating neural word representations in tensor-based compositional settings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 708–719, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1079. URL http://www.aclweb.org/anthology/D14-1079.

Dmitrijs Milajevs, Mehrnoosh Sadrzadeh, and Thomas Roelleke. IR meets NLP: On the semantic similarity between subject-verb-object phrases. In *Proceedings of the 2015 International Conference on Theory of Information Retrieval*, ICTIR '15, pages 231–240, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3833-2. doi: 10.1145/2808194.2809448. URL http://www.eecs.qmul.ac.uk/~dm303/static/ictir006-milajevs.pdf.

Dmitrijs Milajevs, Mehrnoosh Sadrzadeh, and Matthew Purver. Robust co-occurrence quantification for lexical distributional semantics. In *Proceedings of the ACL 2016 Student Research Workshop*, pages 58–64, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-3009. URL http://anthology.aclweb.org/P16-3009.

George A. Miller. Wordnet: A lexical database for English. *Commun. ACM*, 38(11):39–41, November 1995. ISSN 0001-0782. doi: 10.1145/219717.219748. URL http://doi.acm.org/10.1145/219717.219748.

Jeff Mitchell and Mirella Lapata. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P/P08/P08-1028.pdf.

Jeff Mitchell and Mirella Lapata. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429, 2010. ISSN 1551-6709. doi: 10.1111/j.1551-6709.2010.01106.x. URL http://dx.doi.org/10.1111/j.1551-6709.2010.01106.x.

Richard Montague. Universal grammar. *Theoria*, 36(3):373–398, 1970. ISSN 1755-2567. doi: 10.1111/j.1755-2567.1970.tb00434.x. URL http://dx.doi.org/10.1111/j.1755-2567.1970.tb00434.x.

H. Ney, S. Martin, and F. Wessel. *Statistical Language Modeling Using Leaving-One-Out*, pages 174–207. Springer Netherlands, Dordrecht, 1997. ISBN 978-94-017-1183-8. doi: 10.1007/978-94-017-1183-8_6. URL http://dx.doi.org/10.1007/978-94-017-1183-8_6.

Francisco Pereira, Samuel Gershman, Samuel Ritter, and Matthew Botvinick. A comparative evaluation of off-the-shelf distributed semantic representations for modelling behavioural data. *Cognitive Neuropsychology*, 33(3-4):175–190, 2016. doi: 10.1080/02643294.2016.1176907. URL http://dx.doi.org/10.1080/02643294.2016.1176907.

Tamara Polajnar and Stephen Clark. Improving distributional semantic vectors through context selection and normalisation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 230–238, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/E14-1025.

Tamara Polajnar, Laura Rimell, and Stephen Clark. An exploration of discourse-based sentence spaces for compositional distributional semantics. In *Proceedings of the First Workshop on Linking Computational Models of Lexical, Sentential and Discourse-level Semantics*, pages 1–11, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL http://aclweb.org/anthology/W15-2701.

Pushpendre Rastogi, Benjamin Van Durme, and Raman Arora. Multiview lsa: Representation learning via generalized cca. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 556–566, Denver, Colorado, May–June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N15-1058.

Gabriel Recchia and Paul Nulty. Improving a fundamental measure of lexical association. 2017.

Gábor Recski, Eszter Iklódi, Katalin Pajkossy, and Andras Kornai. Measuring semantic similarity of words using concept networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 193–200, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://anthology.aclweb.org/W16-1622.

Herbert Rubenstein and John B. Goodenough. Contextual correlates of synonymy. *Commun. ACM*, 8(10):627–633, October 1965. ISSN 0001-0782. doi: 10.1145/365628.365657. URL http://doi.acm.org/10.1145/365628.365657.

James H Steiger. Tests for comparing elements of a correlation matrix. *Psychological bulletin*, 87(2):245, 1980.

Andreas Stolcke, Noah Coccaro, Rebecca Bates, Paul Taylor, Carol Van Ess-Dykema, Klaus Ries, Elizabeth Shriberg, Daniel Jurafsky, Rachel Martin, and Marie Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Comput. Linguist.*, 26(3): 339–373, September 2000. ISSN 0891-2017. doi: 10.1162/089120100561737. URL http://dx.doi.org/10.1162/089120100561737.

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. Contextualizing semantic representations using syntactically enriched vector models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 948–957, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1858681.1858778.

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. Word meaning in context: A simple and effective vector model. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1134–1143, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing. URL http://www.aclweb.org/anthology/I11-1127.

John C Turner, Michael A Hogg, Penelope J Oakes, Stephen D Reicher, and Margaret S Wetherell. *Rediscovering the social group: A self-categorization theory.* Basil Blackwell, 1987. URL http://psycnet.apa.org/psycinfo/1987-98657-000.

Peter D. Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *J. Artif. Int. Res.*, 37(1):141–188, January 2010. ISSN 1076-9757. URL http://arxiv.org/pdf/1003.1141v1.pdf.

Amos Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977. doi: 10.1037/0033-295x.84.4.327. URL http://dx.doi.org/10.1037/0033-295X.84.4.327.

Amos Tversky and J. Wesley Hutchinson. Nearest neighbor analysis of psychological spaces. *Psychological Review*, 93(1):3 – 22, 1986. ISSN 0033-295X. URL http://search.ebscohost.com/login.aspx?direct=true&db=pdh&AN=1986-13502-001&site=ehost-live.

Justin Washtell and Katja Markert. A comparison of windowless and window-based computational association measures as predictors of syntagmatic human associations. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 628–637, Singapore, August 2009. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D/D09/D09-1066.

Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '96, pages 4–11, New York, NY, USA, 1996. ACM. ISBN 0-89791-792-8. doi: 10.1145/243199.243202. URL http://doi.acm.org/10.1145/243199.243202.

Maayan Zhitomirsky-Geffet and Ido Dagan. Bootstrapping distributional feature vector quality. *Computational Linguistics*, 35(3):435–461, sep 2009. doi: 10.1162/coli.08-032-r1-06-96. URL http://dx.doi.org/10.1162/coli.08-032-R1-06-96.

Aleksandrs Čaks. *Dzejas izlase*. Skolas bibliotēka. Zvaigzne ABC, 1996. ISBN 9789984044101. URL https://books.google.com/books?id=9MXqAAAAMAAJ.

Aleksandrs Čaks and Inara Cendris. *Between Two Rains*. Amazon Digital Services LLC, 2013. URL https://www.amazon.com/gp/product/B00C10SNZG.

# Appendix A

# Experimental data

| operator | dimensionality | KS14 | freq | discr | cds | neg | similarity |
|---|---:|---|---|---|---|---|---|
| head | 1 000 | **0.70** | logn | scpmi | 1 | 0.7 | correlation |
| head | 2 000 | **0.72** | logn | scpmi | 1 | 0.7 | cos |
| head | 3 000 | **0.71** | logn | scpmi | global | 1.4 | cos |
| head | 5 000 | **0.69** | logn | scpmi | 0.75 | 0.7 | inner_product |
| head | 10 000 | **0.71** | 1 | scpmi | 0.75 | 1 | cos |
| head | 20 000 | **0.72** | logn | scpmi | 0.75 | 1 | cos |
| head | 30 000 | **0.72** | logn | scpmi | 0.75 | 1 | cos |
| head | 40 000 | **0.72** | logn | spmi | 0.75 | 1.4 | cos |
| head | 50 000 | **0.72** | logn | spmi | 0.75 | 1.4 | correlation |
| add | 1 000 | **0.78** | 1 | spmi | global | 1.4 | correlation |
| add | 2 000 | **0.80** | 1 | spmi | global | 1 | correlation |
| add | 3 000 | **0.79** | 1 | spmi | global | 1 | correlation |
| add | 5 000 | **0.79** | 1 | scpmi | 0.75 | 0.7 | correlation |
| add | 10 000 | **0.79** | 1 | spmi | 0.75 | 0.5 | correlation |
| add | 20 000 | **0.80** | 1 | scpmi | 0.75 | 0.7 | correlation |
| add | 30 000 | **0.79** | 1 | scpmi | 0.75 | 0.7 | correlation |
| add | 40 000 | **0.79** | 1 | scpmi | 0.75 | 0.7 | correlation |
| add | 50 000 | **0.79** | 1 | scpmi | 0.75 | 0.7 | correlation |
| mult | 1 000 | **0.77** | 1 | scpmi | 1 | 0.2 | correlation |
| mult | 2 000 | **0.78** | 1 | scpmi | 1 | 0.2 | correlation |
| mult | 3 000 | **0.78** | 1 | scpmi | 1 | 0.2 | correlation |
| mult | 5 000 | **0.78** | 1 | scpmi | 1 | 0.2 | correlation |
| mult | 10 000 | **0.78** | 1 | scpmi | 1 | 0.2 | correlation |
| mult | 20 000 | **0.78** | 1 | scpmi | 1 | 0.2 | correlation |
| mult | 30 000 | **0.77** | 1 | scpmi | 1 | 0.2 | correlation |
| mult | 40 000 | **0.77** | 1 | scpmi | 1 | 0.2 | correlation |
| mult | 50 000 | **0.77** | 1 | cpmi | 1 | N/A | correlation |
| kron | 1 000 | **0.80** | 1 | scpmi | 1 | 0.2 | correlation |
| kron | 2 000 | **0.81** | 1 | scpmi | 1 | 0.2 | correlation |
| kron | 3 000 | **0.81** | 1 | scpmi | 1 | 0.2 | correlation |
| kron | 5 000 | **0.79** | 1 | scpmi | 0.75 | 0.7 | inner_product |
| kron | 10 000 | **0.80** | 1 | scpmi | 0.75 | 0.7 | inner_product |
| kron | 20 000 | **0.79** | 1 | scpmi | 0.75 | 0.7 | inner_product |
| kron | 30 000 | **0.79** | 1 | scpmi | 0.75 | 0.7 | inner_product |
| kron | 40 000 | **0.79** | 1 | scpmi | 0.75 | 0.7 | inner_product |
| kron | 50 000 | **0.79** | 1 | spmi | 0.75 | 0.7 | inner_product |

**Table A.1:** KS14 Max selection

| operator | dimensionality | KS14 | freq | discr | cds | neg | similarity |
|---|---:|---|---|---|---|---|---|
| head | 1 000 | **0.69** | logn | scpmi | global | 1 | cos |
| head | 2 000 | **0.70** | logn | scpmi | global | 1 | cos |
| head | 3 000 | **0.69** | logn | scpmi | global | 1 | cos |
| head | 5 000 | **0.69** | logn | scpmi | global | 1.4 | cos |
| head | 10 000 | **0.69** | logn | scpmi | global | 1.4 | cos |
| head | 20 000 | **0.70** | logn | spmi | 0.75 | 1.4 | correlation |
| head | 30 000 | **0.69** | logn | spmi | 0.75 | 2 | correlation |
| head | 40 000 | **0.69** | logn | spmi | 0.75 | 2 | correlation |
| head | 50 000 | **0.69** | logn | spmi | 0.75 | 2 | correlation |
| add | 1 000 | **0.78** | 1 | spmi | global | 1.4 | correlation |
| add | 2 000 | **0.79** | 1 | spmi | global | 1.4 | correlation |
| add | 3 000 | **0.79** | 1 | spmi | global | 1.4 | correlation |
| add | 5 000 | **0.79** | logn | spmi | global | 1.4 | correlation |
| add | 10 000 | **0.79** | logn | spmi | global | 1.4 | correlation |
| add | 20 000 | **0.78** | logn | spmi | global | 2 | correlation |
| add | 30 000 | **0.78** | logn | spmi | global | 2 | correlation |
| add | 40 000 | **0.78** | logn | spmi | global | 2 | correlation |
| add | 50 000 | **0.77** | logn | spmi | global | 2 | correlation |
| mult | 1 000 | **0.77** | 1 | scpmi | global | 0.5 | correlation |
| mult | 2 000 | **0.78** | 1 | scpmi | global | 0.5 | correlation |
| mult | 3 000 | **0.77** | 1 | scpmi | global | 0.5 | correlation |
| mult | 5 000 | **0.77** | 1 | scpmi | global | 0.5 | correlation |
| mult | 10 000 | **0.77** | 1 | scpmi | global | 0.7 | correlation |
| mult | 20 000 | **0.77** | 1 | scpmi | global | 0.7 | correlation |
| mult | 30 000 | **0.75** | 1 | scpmi | global | 1 | correlation |
| mult | 40 000 | **0.74** | 1 | scpmi | global | 1 | correlation |
| mult | 50 000 | **0.74** | 1 | scpmi | global | 1 | correlation |
| kron | 1 000 | **0.78** | 1 | spmi | global | 0.5 | correlation |
| kron | 2 000 | **0.80** | 1 | spmi | global | 0.5 | correlation |
| kron | 3 000 | **0.80** | 1 | spmi | global | 0.7 | correlation |
| kron | 5 000 | **0.77** | 1 | spmi | global | 0.7 | inner_product |
| kron | 10 000 | **0.75** | logn | spmi | global | 0.7 | inner_product |
| kron | 20 000 | **0.77** | logn | scpmi | global | 1 | inner_product |
| kron | 30 000 | **0.77** | logn | scpmi | global | 1 | inner_product |
| kron | 40 000 | **0.77** | logn | scpmi | global | 1 | inner_product |
| kron | 50 000 | **0.77** | logn | scpmi | global | 1 | inner_product |

**Table A.2:** KS14 selection based on heuristics

| operator | dimensionality | GS11 | freq | discr | cds | neg | similarity |
|---|---:|---:|---|---|---|---|---|
| head | 1 000 | **0.38** | logn | scpmi | 0.75 | 0.2 | correlation |
| head | 2 000 | **0.36** | 1 | pmi | global | N/A | inner_product |
| head | 3 000 | **0.41** | 1 | pmi | global | N/A | inner_product |
| head | 5 000 | **0.40** | 1 | pmi | global | N/A | inner_product |
| head | 10 000 | **0.43** | 1 | pmi | global | N/A | inner_product |
| head | 20 000 | **0.37** | 1 | scpmi | global | 1 | correlation |
| head | 30 000 | **0.38** | logn | spmi | global | 0.7 | correlation |
| head | 40 000 | **0.38** | logn | scpmi | 1 | 0.7 | correlation |
| head | 50 000 | **0.38** | logn | spmi | global | 0.7 | correlation |
| add | 1 000 | **0.34** | 1 | scpmi | global | 0.7 | correlation |
| add | 2 000 | **0.31** | 1 | spmi | global | 0.2 | correlation |
| add | 3 000 | **0.31** | 1 | pmi | 0.75 | N/A | correlation |
| add | 5 000 | **0.30** | 1 | pmi | 1 | N/A | cos |
| add | 10 000 | **0.32** | 1 | pmi | global | N/A | cos |
| add | 20 000 | **0.28** | logn | scpmi | 0.75 | 0.2 | correlation |
| add | 30 000 | 0.27 | logn | scpmi | 0.75 | 0.2 | correlation |
| add | 40 000 | 0.26 | 1 | pmi | global | N/A | correlation |
| add | 50 000 | 0.25 | 1 | pmi | global | N/A | correlation |
| mult | 1 000 | 0.46 | logn | pmi | global | N/A | inner_product |
| mult | 2 000 | 0.47 | logn | spmi | global | 0.5 | cos |
| mult | 3 000 | **0.48** | 1 | scpmi | global | 0.7 | cos |
| mult | 5 000 | **0.49** | 1 | scpmi | global | 0.7 | cos |
| mult | 10 000 | **0.50** | logn | spmi | global | 0.5 | cos |
| mult | 20 000 | **0.53** | 1 | scpmi | global | 0.7 | correlation |
| mult | 30 000 | **0.51** | 1 | scpmi | global | 1 | correlation |
| mult | 40 000 | **0.52** | 1 | scpmi | global | 1 | correlation |
| mult | 50 000 | **0.50** | logn | spmi | global | 0.5 | cos |
| kron | 1 000 | 0.43 | 1 | scpmi | global | 0.2 | correlation |
| kron | 2 000 | **0.49** | logn | scpmi | 0.75 | 0.7 | inner_product |
| kron | 3 000 | **0.50** | logn | spmi | 0.75 | 0.5 | inner_product |
| kron | 5 000 | **0.51** | logn | spmi | 0.75 | 0.7 | inner_product |
| kron | 10 000 | **0.51** | logn | spmi | 0.75 | 0.7 | inner_product |
| kron | 20 000 | **0.51** | logn | spmi | 0.75 | 0.7 | inner_product |
| kron | 30 000 | **0.51** | logn | spmi | 0.75 | 1 | inner_product |
| kron | 40 000 | **0.51** | logn | spmi | 0.75 | 1 | inner_product |
| kron | 50 000 | **0.52** | logn | spmi | 0.75 | 1 | inner_product |

**Table A.3:** GS11 Max selection

| operator | dimensionality | GS11 | freq | discr | cds | neg | similarity |
|---|---:|---:|---|---|---|---|---|
| head | 1 000 | **0.33** | logn | spmi | global | 0.5 | cos |
| head | 2 000 | **0.33** | logn | spmi | global | 0.5 | cos |
| head | 3 000 | **0.35** | logn | spmi | global | 0.7 | cos |
| head | 5 000 | **0.36** | logn | spmi | global | 0.7 | cos |
| head | 10 000 | **0.34** | logn | spmi | 1 | 0.7 | cos |
| head | 20 000 | **0.36** | logn | spmi | 1 | 0.7 | cos |
| head | 30 000 | **0.36** | logn | spmi | 1 | 0.7 | cos |
| head | 40 000 | **0.37** | logn | spmi | 1 | 0.7 | cos |
| head | 50 000 | **0.36** | logn | spmi | 1 | 0.7 | cos |
| add | 1 000 | **0.29** | logn | pmi | 1 | N/A | correlation |
| add | 2 000 | **0.28** | logn | pmi | 1 | N/A | correlation |
| add | 3 000 | **0.28** | logn | pmi | 1 | N/A | correlation |
| add | 5 000 | **0.28** | logn | pmi | 1 | N/A | correlation |
| add | 10 000 | **0.26** | logn | pmi | 1 | N/A | correlation |
| add | 20 000 | **0.28** | logn | scpmi | 0.75 | 0.2 | correlation |
| add | 30 000 | **0.27** | logn | scpmi | 0.75 | 0.2 | correlation |
| add | 40 000 | **0.25** | logn | scpmi | 0.75 | 0.2 | correlation |
| add | 50 000 | **0.24** | logn | scpmi | 0.75 | 0.2 | correlation |
| mult | 1 000 | 0.44 | logn | pmi | global | N/A | cos |
| mult | 2 000 | 0.41 | logn | pmi | global | N/A | cos |
| mult | 3 000 | 0.44 | logn | pmi | global | N/A | cos |
| mult | 5 000 | **0.49** | logn | scpmi | global | 0.7 | cos |
| mult | 10 000 | **0.50** | logn | scpmi | global | 0.7 | cos |
| mult | 20 000 | **0.50** | logn | scpmi | global | 0.7 | cos |
| mult | 30 000 | **0.50** | logn | scpmi | global | 0.7 | cos |
| mult | 40 000 | **0.51** | logn | scpmi | global | 0.7 | cos |
| mult | 50 000 | **0.50** | logn | scpmi | global | 0.7 | cos |
| kron | 1 000 | 0.41 | logn | scpmi | 0.75 | 0.7 | inner_product |
| kron | 2 000 | **0.49** | logn | scpmi | 0.75 | 0.7 | inner_product |
| kron | 3 000 | **0.50** | logn | scpmi | 0.75 | 0.7 | inner_product |
| kron | 5 000 | **0.51** | logn | spmi | 0.75 | 0.7 | inner_product |
| kron | 10 000 | **0.51** | logn | spmi | 0.75 | 0.7 | inner_product |
| kron | 20 000 | **0.51** | logn | spmi | 0.75 | 1 | inner_product |
| kron | 30 000 | **0.51** | logn | spmi | 0.75 | 1 | inner_product |
| kron | 40 000 | **0.51** | logn | spmi | 0.75 | 1 | inner_product |
| kron | 50 000 | **0.52** | logn | spmi | 0.75 | 1 | inner_product |

**Table A.4:** GS11 selection based on heuristics

| operator | dimensionality | PhraseRel | freq | discr | cds | neg | similarity |
|---|---:|---:|---|---|---|---|---|
| head | 1 000 | **0.71** | 1 | spmi | 0.75 | 0.7 | correlation |
| head | 2 000 | **0.75** | n | spmi | 0.75 | 1.4 | correlation |
| head | 3 000 | **0.75** | logn | spmi | 0.75 | 2 | inner_product |
| head | 5 000 | **0.75** | n | spmi | 0.75 | 2 | correlation |
| head | 10 000 | **0.75** | 1 | scpmi | 0.75 | 2 | correlation |
| head | 20 000 | **0.75** | 1 | spmi | global | 0.2 | inner_product |
| head | 30 000 | **0.75** | n | scpmi | global | 1.4 | cos |
| head | 40 000 | **0.75** | n | scpmi | global | 1.4 | correlation |
| head | 50 000 | **0.75** | 1 | spmi | 1 | 2 | cos |
| add | 1 000 | **0.89** | 1 | pmi | global | N/A | cos |
| add | 2 000 | **0.89** | 1 | cpmi | 1 | N/A | correlation |
| add | 3 000 | **0.86** | 1 | spmi | 0.75 | 0.7 | correlation |
| add | 5 000 | **0.89** | n | spmi | 0.75 | 5 | correlation |
| add | 10 000 | **0.86** | 1 | spmi | 1 | 0.2 | inner_product |
| add | 20 000 | **0.86** | 1 | spmi | 0.75 | 0.2 | inner_product |
| add | 30 000 | **0.89** | n | spmi | 0.75 | 7 | correlation |
| add | 40 000 | **0.86** | 1 | spmi | 0.75 | 0.5 | inner_product |
| add | 50 000 | **0.86** | 1 | spmi | 0.75 | 0.5 | inner_product |
| mult | 1 000 | **0.93** | 1 | cpmi | 0.75 | N/A | correlation |
| mult | 2 000 | **0.96** | logn | spmi | 0.75 | 0.2 | correlation |
| mult | 3 000 | **0.93** | 1 | cpmi | 0.75 | N/A | cos |
| mult | 5 000 | **0.96** | logn | spmi | 0.75 | 0.2 | correlation |
| mult | 10 000 | **1.00** | 1 | scpmi | 1 | 0.7 | correlation |
| mult | 20 000 | **1.00** | logn | cpmi | 1 | N/A | correlation |
| mult | 30 000 | **0.96** | logn | cpmi | 1 | N/A | correlation |
| mult | 40 000 | **0.96** | logn | cpmi | 1 | N/A | correlation |
| mult | 50 000 | **0.96** | logn | cpmi | 1 | N/A | correlation |
| kron | 1 000 | **0.93** | 1 | scpmi | global | 0.2 | correlation |
| kron | 2 000 | **0.93** | 1 | cpmi | 0.75 | N/A | correlation |
| kron | 3 000 | **0.96** | 1 | scpmi | global | 1 | correlation |
| kron | 5 000 | **0.89** | 1 | scpmi | 0.75 | 1 | inner_product |
| kron | 10 000 | **0.89** | 1 | scpmi | 0.75 | 1 | inner_product |
| kron | 20 000 | **0.89** | 1 | scpmi | 0.75 | 1 | inner_product |
| kron | 30 000 | **0.93** | 1 | spmi | 1 | 1.4 | inner_product |
| kron | 40 000 | **0.93** | 1 | scpmi | global | 2 | inner_product |
| kron | 50 000 | **0.93** | 1 | scpmi | global | 2 | inner_product |

**Table A.5:** PhraseRel Max selection

| operator | dimensionality | PhraseRel | freq | discr | cds | neg | similarity |
|---|---:|---:|---|---|---|---|---|
| head | 1 000 | **0.64** | n | spmi | 0.75 | 1.4 | correlation |
| head | 2 000 | **0.75** | n | spmi | 0.75 | 1.4 | correlation |
| head | 3 000 | **0.71** | n | spmi | 0.75 | 1.4 | correlation |
| head | 5 000 | 0.71 | n | scpmi | 0.75 | 1.4 | correlation |
| head | 10 000 | **0.75** | n | scpmi | global | 1.4 | cos |
| head | 20 000 | **0.75** | n | scpmi | global | 1.4 | cos |
| head | 30 000 | **0.75** | n | scpmi | global | 1.4 | cos |
| head | 40 000 | **0.75** | n | scpmi | global | 1.4 | cos |
| head | 50 000 | **0.75** | n | scpmi | global | 1.4 | cos |
| add | 1 000 | **0.79** | 1 | spmi | global | 2 | cos |
| add | 2 000 | **0.82** | 1 | spmi | global | 2 | cos |
| add | 3 000 | **0.86** | 1 | spmi | global | 2 | cos |
| add | 5 000 | **0.82** | 1 | spmi | global | 2 | cos |
| add | 10 000 | **0.82** | 1 | spmi | global | 2 | cos |
| add | 20 000 | **0.82** | 1 | spmi | global | 2 | inner_product |
| add | 30 000 | **0.82** | 1 | spmi | global | 2 | inner_product |
| add | 40 000 | **0.79** | 1 | spmi | global | 2 | inner_product |
| add | 50 000 | **0.75** | 1 | spmi | global | 2 | inner_product |
| mult | 1 000 | **0.89** | logn | scpmi | global | 0.5 | correlation |
| mult | 2 000 | **0.89** | logn | scpmi | global | 0.5 | correlation |
| mult | 3 000 | **0.89** | logn | scpmi | global | 0.5 | correlation |
| mult | 5 000 | **0.93** | logn | scpmi | global | 0.5 | correlation |
| mult | 10 000 | **0.93** | logn | scpmi | global | 0.5 | correlation |
| mult | 20 000 | **0.96** | logn | scpmi | global | 0.5 | correlation |
| mult | 30 000 | **0.96** | logn | scpmi | global | 0.5 | correlation |
| mult | 40 000 | **0.93** | logn | scpmi | global | 0.5 | correlation |
| mult | 50 000 | **0.93** | logn | scpmi | global | 0.5 | correlation |
| kron | 1 000 | **0.86** | logn | spmi | 1 | 0.5 | correlation |
| kron | 2 000 | **0.93** | logn | spmi | 1 | 0.5 | correlation |
| kron | 3 000 | **0.93** | logn | spmi | 1 | 0.5 | correlation |
| kron | 5 000 | **0.79** | 1 | spmi | 1 | 1 | inner_product |
| kron | 10 000 | **0.86** | 1 | spmi | 1 | 1 | inner_product |
| kron | 20 000 | **0.82** | 1 | spmi | 1 | 1.4 | inner_product |
| kron | 30 000 | **0.93** | 1 | spmi | 1 | 1.4 | inner_product |
| kron | 40 000 | **0.86** | 1 | spmi | 1 | 1.4 | inner_product |
| kron | 50 000 | **0.82** | 1 | spmi | 1 | 1.4 | inner_product |

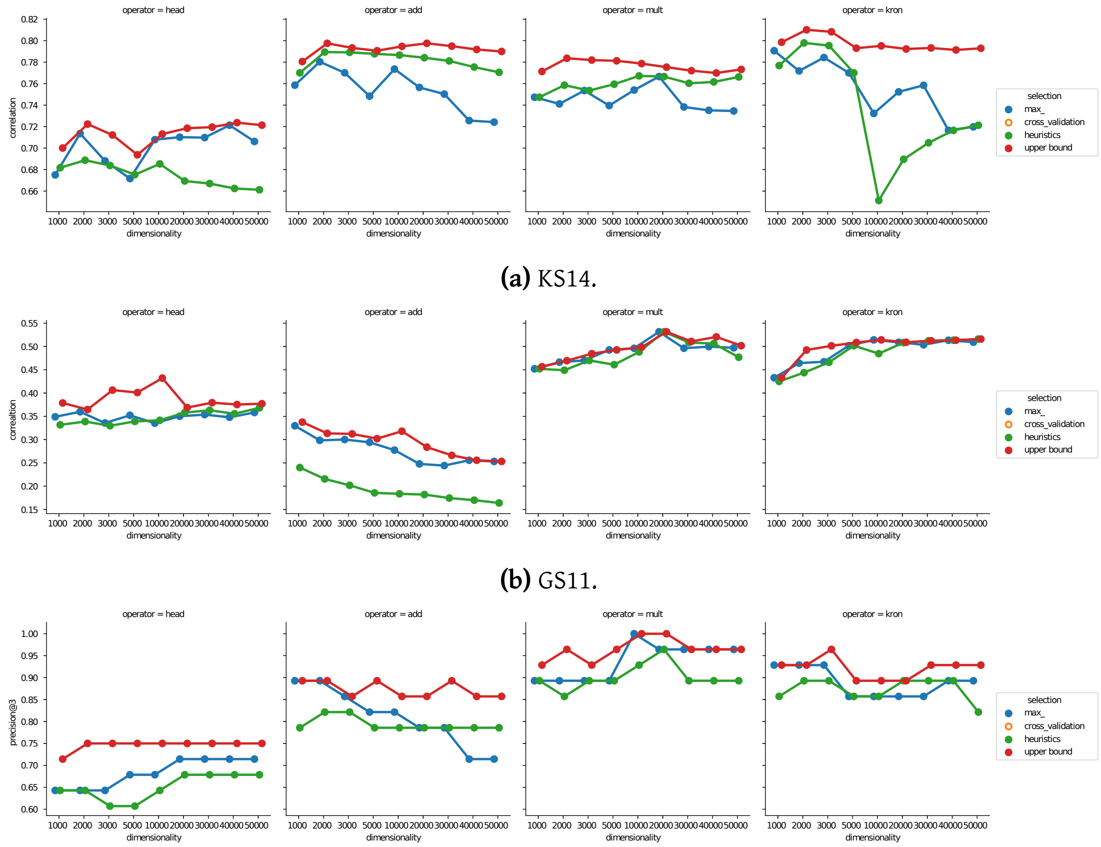
**Table A.6:** PhraseRel selection based on heuristics

| operator | dimensionality | KS14 | GS11 | PhraseRel | compositional | freq | discr | cds | neg | similarity |
|---|---:|---:|---:|---:|---:|---|---|---|---|---|
| head | 1 000 | **0.68** | **0.35** | **0.64** | **0.71** | 1 | spmi | global | 1 | correlation |
| head | 2 000 | **0.71** | **0.36** | **0.64** | **0.73** | logn | spmi | global | 1.4 | inner_product |
| head | 3 000 | **0.69** | **0.34** | **0.64** | **0.71** | logn | spmi | 1 | 0.5 | inner_product |
| head | 5 000 | **0.67** | **0.35** | **0.68** | **0.72** | logn | spmi | 1 | 0.5 | inner_product |
| head | 10 000 | **0.71** | **0.34** | **0.68** | **0.73** | logn | spmi | 0.75 | 1 | cos |
| head | 20 000 | **0.71** | **0.35** | **0.71** | **0.75** | logn | spmi | 0.75 | 1 | cos |
| head | 30 000 | **0.71** | **0.35** | **0.71** | **0.75** | logn | spmi | 0.75 | 1 | cos |
| head | 40 000 | **0.72** | **0.35** | **0.71** | **0.75** | logn | scpmi | 0.75 | 1 | cos |
| head | 50 000 | **0.71** | **0.36** | **0.71** | **0.75** | logn | spmi | 0.75 | 1 | cos |
| add | 1 000 | **0.76** | **0.33** | **0.89** | **0.82** | 1 | spmi | global | 0.5 | correlation |
| add | 2 000 | **0.78** | **0.30** | **0.89** | **0.81** | 1 | scpmi | global | 0.7 | correlation |
| add | 3 000 | **0.77** | **0.30** | **0.86** | **0.79** | 1 | spmi | global | 0.5 | correlation |
| add | 5 000 | **0.75** | **0.29** | **0.82** | **0.77** | logn | scpmi | 0.75 | 0.2 | correlation |
| add | 10 000 | **0.77** | **0.28** | **0.82** | **0.77** | 1 | spmi | 0.75 | 0.2 | correlation |
| add | 20 000 | **0.76** | 0.25 | **0.79** | **0.73** | logn | cpmi | 0.75 | N/A | correlation |
| add | 30 000 | **0.75** | 0.24 | **0.79** | **0.72** | logn | cpmi | 0.75 | N/A | correlation |
| add | 40 000 | **0.73** | 0.26 | **0.71** | **0.70** | 1 | pmi | global | N/A | correlation |
| add | 50 000 | **0.72** | 0.25 | **0.71** | **0.69** | 1 | pmi | global | N/A | correlation |
| mult | 1 000 | **0.75** | 0.45 | **0.89** | **0.89** | 1 | spmi | global | 0.5 | correlation |
| mult | 2 000 | **0.74** | 0.47 | **0.89** | **0.90** | logn | spmi | global | 0.5 | correlation |
| mult | 3 000 | **0.75** | 0.47 | **0.89** | **0.90** | 1 | spmi | global | 0.5 | correlation |
| mult | 5 000 | **0.74** | **0.49** | **0.89** | **0.91** | 1 | scpmi | global | 0.7 | cos |
| mult | 10 000 | **0.75** | **0.50** | **1.00** | **0.95** | logn | spmi | global | 0.5 | correlation |
| mult | 20 000 | **0.77** | **0.53** | **0.96** | **0.97** | 1 | scpmi | global | 0.7 | correlation |
| mult | 30 000 | **0.74** | **0.50** | **0.96** | **0.94** | logn | spmi | 1 | 0.2 | correlation |
| mult | 40 000 | **0.74** | **0.50** | **0.96** | **0.94** | logn | spmi | 1 | 0.2 | cos |
| mult | 50 000 | **0.73** | **0.50** | **0.96** | **0.94** | logn | spmi | 1 | 0.2 | correlation |
| kron | 1 000 | **0.79** | 0.43 | **0.93** | **0.91** | 1 | scpmi | global | 0.2 | correlation |
| kron | 2 000 | **0.77** | **0.46** | **0.93** | **0.92** | 1 | spmi | 0.75 | 0.2 | correlation |
| kron | 3 000 | **0.78** | **0.47** | **0.93** | **0.93** | 1 | scpmi | global | 0.7 | cos |
| kron | 5 000 | **0.77** | **0.50** | **0.86** | **0.92** | logn | scpmi | 0.75 | 0.7 | inner_product |
| kron | 10 000 | **0.73** | **0.51** | **0.86** | **0.91** | logn | spmi | 0.75 | 0.7 | inner_product |
| kron | 20 000 | **0.75** | **0.51** | **0.86** | **0.91** | logn | spmi | 0.75 | 0.7 | inner_product |
| kron | 30 000 | **0.76** | **0.50** | **0.86** | **0.91** | logn | spmi | 0.75 | 0.7 | inner_product |
| kron | 40 000 | **0.72** | **0.51** | **0.89** | **0.91** | logn | spmi | 0.75 | 1 | inner_product |
| kron | 50 000 | **0.72** | **0.51** | **0.89** | **0.91** | logn | scpmi | 0.75 | 1 | inner_product |

**Table A.7:** Compositional (combined KS13, GS11 and PhraseRel) Max selection

| operator | dimensionality | KS14 | GS11 | PhraseRel | compositional | freq | discr | cds | neg | similarity |
|---|---|---|---|---|---|---|---|---|---|---|
| head | 1 000 | **0.68** | **0.33** | **0.64** | **0.70** | logn | spmi | global | 1 | correlation |
| head | 2 000 | **0.69** | **0.34** | **0.64** | **0.71** | logn | spmi | global | 1 | correlation |
| head | 3 000 | **0.68** | **0.33** | **0.61** | **0.69** | logn | spmi | global | 1 | correlation |
| head | 5 000 | **0.68** | **0.34** | **0.61** | **0.69** | logn | spmi | global | 1 | cos |
| head | 10 000 | **0.69** | **0.34** | **0.64** | **0.71** | logn | spmi | global | 1.4 | cos |
| head | 20 000 | **0.67** | **0.36** | **0.68** | **0.73** | logn | spmi | global | 1.4 | cos |
| head | 30 000 | **0.67** | **0.36** | **0.68** | **0.73** | logn | spmi | global | 1.4 | cos |
| head | 40 000 | **0.66** | **0.36** | **0.68** | **0.72** | logn | spmi | global | 1.4 | cos |
| head | 50 000 | **0.66** | **0.37** | **0.68** | **0.73** | logn | spmi | global | 1.4 | cos |
| add | 1 000 | **0.77** | **0.24** | **0.79** | **0.73** | logn | spmi | global | 1 | correlation |
| add | 2 000 | **0.79** | **0.22** | **0.82** | **0.73** | logn | spmi | global | 1 | correlation |
| add | 3 000 | **0.79** | **0.20** | **0.82** | **0.73** | logn | spmi | global | 1 | correlation |
| add | 5 000 | **0.79** | **0.19** | **0.79** | **0.70** | logn | spmi | global | 1 | correlation |
| add | 10 000 | **0.79** | 0.18 | **0.79** | **0.70** | logn | spmi | global | 1 | correlation |
| add | 20 000 | **0.78** | 0.18 | **0.79** | **0.70** | logn | spmi | global | 1 | correlation |
| add | 30 000 | **0.78** | 0.17 | **0.79** | **0.69** | logn | spmi | global | 1 | correlation |
| add | 40 000 | **0.78** | 0.17 | **0.79** | **0.69** | logn | spmi | global | 1 | correlation |
| add | 50 000 | **0.77** | 0.16 | **0.79** | **0.68** | logn | spmi | global | 1 | correlation |
| mult | 1 000 | **0.75** | 0.45 | **0.89** | **0.89** | 1 | spmi | global | 0.5 | correlation |
| mult | 2 000 | **0.76** | 0.45 | **0.86** | **0.88** | 1 | spmi | global | 0.5 | correlation |
| mult | 3 000 | **0.75** | 0.47 | **0.89** | **0.90** | 1 | spmi | global | 0.5 | correlation |
| mult | 5 000 | **0.76** | 0.46 | **0.89** | **0.90** | 1 | spmi | global | 0.5 | correlation |
| mult | 10 000 | **0.77** | **0.49** | **0.93** | **0.93** | 1 | scpmi | global | 0.7 | correlation |
| mult | 20 000 | **0.77** | **0.53** | **0.96** | **0.97** | 1 | scpmi | global | 0.7 | correlation |
| mult | 30 000 | **0.76** | **0.51** | **0.89** | **0.93** | 1 | scpmi | global | 0.7 | correlation |
| mult | 40 000 | **0.76** | **0.51** | **0.89** | **0.93** | 1 | scpmi | global | 0.7 | correlation |
| mult | 50 000 | **0.77** | **0.48** | **0.89** | **0.91** | 1 | scpmi | global | 0.7 | correlation |
| kron | 1 000 | **0.78** | 0.42 | **0.86** | **0.87** | 1 | scpmi | global | 0.7 | correlation |
| kron | 2 000 | **0.80** | 0.44 | **0.89** | **0.90** | 1 | scpmi | global | 0.7 | correlation |
| kron | 3 000 | **0.80** | **0.47** | **0.89** | **0.92** | 1 | scpmi | global | 0.7 | correlation |
| kron | 5 000 | **0.77** | **0.50** | **0.86** | **0.92** | logn | scpmi | 0.75 | 0.7 | inner_product |
| kron | 10 000 | 0.65 | **0.48** | **0.86** | **0.86** | logn | spmi | 0.75 | 1 | inner_product |
| kron | 20 000 | 0.69 | **0.51** | **0.89** | **0.90** | logn | spmi | 0.75 | 1 | inner_product |
| kron | 30 000 | 0.70 | **0.51** | **0.89** | **0.91** | logn | spmi | 0.75 | 1 | inner_product |
| kron | 40 000 | 0.72 | **0.51** | **0.89** | **0.91** | logn | spmi | 0.75 | 1 | inner_product |
| kron | 50 000 | 0.72 | **0.52** | **0.82** | **0.89** | logn | spmi | 0.75 | 1 | inner_product |

**Table A.8:** Compositional (combined KS13, GS11 and PhraseRel) selection based on heuristics



**(a)** KS14.

**(b)** GS11.

**(c)** PhraseRel.

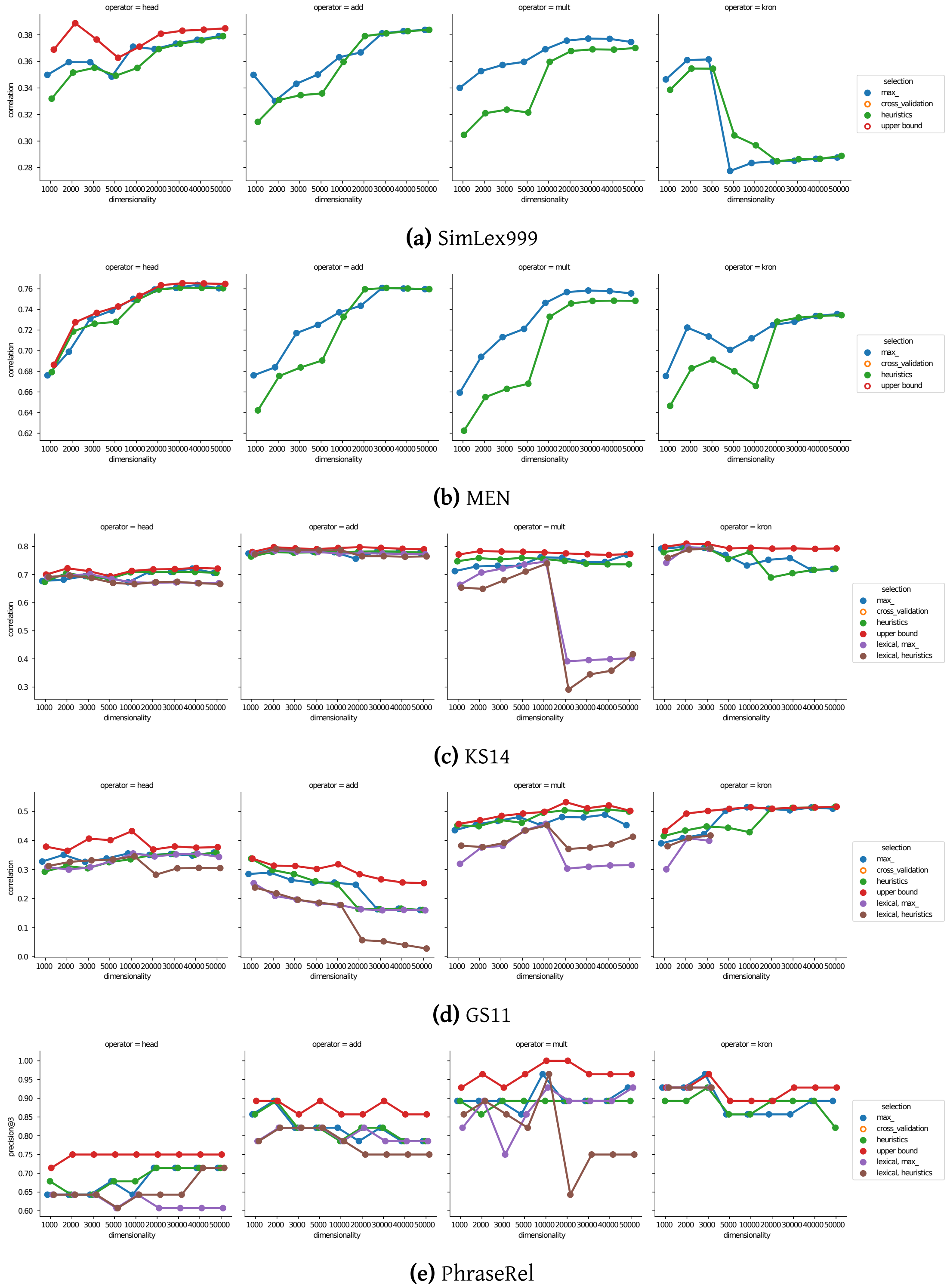
**Figure A.1:** Performance of models based on the selection over the average compositional performance

| operator | dimensionality | SimLex999 | men | KS14 | GS11 | PhraseRel | universal | freq | discr | cds | neg | similarity |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| head | 1 000 | 0.35 | 0.68 | **0.68** | **0.33** | **0.64** | **0.79** | 1 | scpmi | global | 1 | cos |
| head | 2 000 | **0.36** | 0.70 | **0.68** | **0.35** | **0.64** | **0.82** | 1 | spmi | global | 1 | cos |
| head | 3 000 | **0.36** | 0.73 | **0.69** | **0.33** | **0.64** | **0.82** | logn | scpmi | global | 1 | cos |
| head | 5 000 | 0.35 | 0.74 | **0.69** | **0.34** | **0.68** | **0.83** | logn | spmi | 0.75 | 0.7 | cos |
| head | 10 000 | **0.37** | **0.75** | **0.67** | **0.36** | **0.64** | **0.84** | logn | scpmi | 1 | 0.7 | cos |
| head | 20 000 | **0.37** | **0.76** | **0.71** | **0.35** | **0.71** | **0.86** | logn | spmi | 0.75 | 1 | cos |
| head | 30 000 | **0.37** | **0.76** | **0.71** | **0.35** | **0.71** | **0.86** | logn | spmi | 0.75 | 1 | cos |
| head | 40 000 | **0.38** | **0.76** | **0.72** | **0.35** | **0.71** | **0.87** | logn | scpmi | 0.75 | 1 | cos |
| head | 50 000 | **0.38** | **0.76** | **0.71** | **0.36** | **0.71** | **0.87** | logn | spmi | 0.75 | 1 | cos |
| add | 1 000 | 0.35 | 0.68 | **0.77** | **0.28** | **0.86** | **0.84** | 1 | scpmi | global | 1 | cos |
| add | 2 000 | 0.33 | 0.68 | **0.79** | **0.29** | **0.89** | **0.84** | 1 | cpmi | 1 | N/A | correlation |
| add | 3 000 | 0.34 | 0.72 | **0.78** | **0.26** | **0.82** | **0.84** | logn | cpmi | 1 | N/A | correlation |
| add | 5 000 | 0.35 | 0.73 | **0.78** | **0.25** | **0.82** | **0.84** | logn | cpmi | 1 | N/A | correlation |
| add | 10 000 | **0.36** | 0.74 | **0.78** | **0.26** | **0.82** | **0.85** | logn | cpmi | 1 | N/A | correlation |
| add | 20 000 | **0.37** | 0.74 | **0.76** | **0.25** | **0.79** | **0.84** | logn | cpmi | 0.75 | N/A | correlation |
| add | 30 000 | **0.38** | **0.76** | **0.78** | 0.16 | 0.82 | **0.84** | logn | scpmi | 0.75 | 0.7 | correlation |
| add | 40 000 | **0.38** | **0.76** | **0.78** | 0.17 | 0.79 | **0.84** | logn | scpmi | 0.75 | 0.7 | correlation |
| add | 50 000 | **0.38** | **0.76** | **0.78** | 0.16 | 0.79 | **0.84** | logn | scpmi | 0.75 | 0.7 | correlation |
| mult | 1 000 | 0.34 | 0.66 | **0.71** | **0.44** | **0.89** | **0.87** | logn | spmi | global | 0.7 | cos |
| mult | 2 000 | 0.35 | 0.69 | **0.73** | **0.46** | **0.89** | **0.89** | logn | spmi | 1 | 0.2 | cos |
| mult | 3 000 | **0.36** | 0.71 | **0.73** | **0.47** | **0.89** | **0.91** | logn | spmi | global | 0.7 | cos |
| mult | 5 000 | **0.36** | 0.72 | **0.73** | **0.48** | **0.86** | **0.91** | logn | spmi | global | 0.7 | cos |
| mult | 10 000 | **0.37** | **0.75** | **0.76** | **0.45** | **0.96** | **0.94** | logn | scpmi | global | 1 | cos |
| mult | 20 000 | **0.38** | **0.76** | **0.76** | **0.48** | **0.89** | **0.94** | logn | scpmi | global | 1 | cos |
| mult | 30 000 | **0.38** | **0.76** | **0.74** | **0.48** | **0.89** | **0.94** | logn | scpmi | global | 1 | cos |
| mult | 40 000 | **0.38** | **0.76** | **0.74** | **0.49** | **0.89** | **0.95** | logn | scpmi | global | 1 | cos |
| mult | 50 000 | **0.37** | **0.76** | **0.77** | **0.45** | **0.93** | **0.94** | logn | spmi | global | 1.4 | cos |
| kron | 1 000 | **0.35** | 0.68 | **0.79** | 0.39 | **0.93** | **0.88** | logn | spmi | global | 1 | cos |
| kron | 2 000 | **0.36** | 0.72 | **0.80** | 0.41 | **0.93** | **0.91** | logn | scpmi | global | 1 | cos |
| kron | 3 000 | **0.36** | 0.71 | **0.80** | 0.42 | **0.96** | **0.92** | 1 | scpmi | global | 1 | cos |
| kron | 5 000 | 0.28 | 0.70 | **0.77** | **0.50** | **0.86** | **0.87** | logn | scpmi | 0.75 | 0.7 | inner_product |
| kron | 10 000 | 0.28 | 0.71 | **0.73** | **0.51** | **0.86** | **0.87** | logn | spmi | 0.75 | 0.7 | inner_product |
| kron | 20 000 | 0.28 | 0.72 | **0.75** | **0.51** | **0.86** | **0.88** | logn | spmi | 0.75 | 0.7 | inner_product |
| kron | 30 000 | 0.29 | **0.73** | **0.76** | **0.50** | **0.86** | **0.88** | logn | spmi | 0.75 | 0.7 | inner_product |
| kron | 40 000 | 0.29 | **0.73** | 0.72 | **0.51** | **0.89** | **0.88** | logn | spmi | 0.75 | 1 | inner_product |
| kron | 50 000 | 0.29 | **0.74** | 0.72 | **0.51** | **0.89** | **0.88** | logn | scpmi | 0.75 | 1 | inner_product |

**Table A.9:** Universal (operator-dependent) Max selection

| operator | dimensionality | SimLex999 | men | KS14 | GS11 | PhraseRel | universal | freq | discr | cds | neg | similarity |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| head | 1 000 | 0.33 | 0.68 | **0.67** | 0.29 | **0.68** | **0.78** | 1 | scpmi | 0.75 | 0.7 | cos |
| head | 2 000 | 0.35 | 0.72 | **0.70** | **0.31** | **0.64** | **0.81** | 1 | scpmi | 0.75 | 0.7 | cos |
| head | 3 000 | **0.36** | 0.73 | **0.69** | 0.30 | **0.64** | **0.81** | 1 | scpmi | 0.75 | 0.7 | cos |
| head | 5 000 | 0.35 | 0.73 | **0.68** | **0.33** | **0.68** | **0.82** | 1 | spmi | 0.75 | 0.7 | cos |
| head | 10 000 | **0.36** | **0.75** | **0.71** | **0.34** | **0.68** | **0.84** | logn | spmi | 0.75 | 1 | cos |
| head | 20 000 | **0.37** | **0.76** | **0.71** | **0.35** | **0.71** | **0.86** | logn | spmi | 0.75 | 1 | cos |
| head | 30 000 | **0.37** | **0.76** | **0.71** | **0.35** | **0.71** | **0.86** | logn | spmi | 0.75 | 1 | cos |
| head | 40 000 | **0.38** | **0.76** | **0.71** | **0.35** | **0.71** | **0.87** | logn | spmi | 0.75 | 1 | cos |
| head | 50 000 | **0.38** | **0.76** | **0.71** | **0.36** | **0.71** | **0.87** | logn | spmi | 0.75 | 1 | cos |
| add | 1 000 | 0.31 | 0.64 | **0.76** | **0.34** | **0.86** | **0.82** | 1 | scpmi | global | 0.7 | correlation |
| add | 2 000 | 0.33 | 0.68 | **0.78** | **0.30** | **0.89** | **0.84** | 1 | scpmi | global | 0.7 | correlation |
| add | 3 000 | 0.33 | 0.68 | **0.78** | 0.28 | **0.82** | **0.82** | 1 | scpmi | global | 0.7 | correlation |
| add | 5 000 | 0.34 | 0.69 | **0.78** | 0.26 | **0.82** | **0.82** | 1 | scpmi | global | 0.7 | correlation |
| add | 10 000 | **0.36** | 0.73 | **0.78** | 0.25 | **0.79** | **0.84** | logn | scpmi | global | 0.7 | correlation |
| add | 20 000 | **0.38** | **0.76** | **0.78** | 0.16 | **0.82** | **0.84** | logn | scpmi | 0.75 | 0.7 | correlation |
| add | 30 000 | **0.38** | **0.76** | **0.78** | 0.16 | **0.82** | **0.84** | logn | scpmi | 0.75 | 0.7 | correlation |
| add | 40 000 | **0.38** | **0.76** | **0.78** | 0.17 | **0.79** | **0.84** | logn | scpmi | 0.75 | 0.7 | correlation |
| add | 50 000 | **0.38** | **0.76** | **0.78** | 0.16 | **0.79** | **0.84** | logn | scpmi | 0.75 | 0.7 | correlation |
| mult | 1 000 | 0.30 | 0.62 | **0.75** | 0.45 | **0.89** | **0.84** | 1 | spmi | global | 0.5 | correlation |
| mult | 2 000 | 0.32 | 0.66 | **0.76** | 0.45 | **0.86** | **0.86** | 1 | spmi | global | 0.5 | correlation |
| mult | 3 000 | 0.32 | 0.66 | **0.75** | **0.47** | **0.89** | **0.88** | 1 | spmi | global | 0.5 | correlation |
| mult | 5 000 | 0.32 | 0.67 | **0.76** | **0.46** | **0.89** | **0.87** | 1 | spmi | global | 0.5 | correlation |
| mult | 10 000 | **0.36** | 0.73 | **0.76** | **0.50** | **0.89** | **0.93** | logn | scpmi | global | 0.7 | correlation |
| mult | 20 000 | **0.37** | **0.75** | 0.75 | **0.50** | **0.89** | **0.94** | logn | scpmi | global | 0.7 | correlation |
| mult | 30 000 | **0.37** | **0.75** | 0.74 | **0.50** | **0.89** | **0.94** | logn | scpmi | global | 0.7 | correlation |
| mult | 40 000 | **0.37** | **0.75** | 0.74 | **0.51** | **0.89** | **0.94** | logn | scpmi | global | 0.7 | correlation |
| mult | 50 000 | **0.37** | **0.75** | 0.74 | **0.50** | **0.89** | **0.94** | logn | scpmi | global | 0.7 | correlation |
| kron | 1 000 | **0.34** | 0.65 | **0.78** | 0.42 | **0.89** | **0.87** | 1 | spmi | global | 0.7 | cos |
| kron | 2 000 | **0.35** | 0.68 | **0.79** | 0.43 | **0.89** | **0.90** | 1 | spmi | global | 0.7 | cos |
| kron | 3 000 | **0.35** | 0.69 | **0.80** | 0.45 | **0.93** | **0.91** | 1 | spmi | global | 0.7 | cos |
| kron | 5 000 | 0.30 | 0.68 | **0.76** | 0.44 | **0.86** | **0.86** | 1 | spmi | 0.75 | 0.7 | inner_product |
| kron | 10 000 | 0.30 | 0.67 | **0.78** | 0.43 | **0.86** | **0.85** | 1 | spmi | 0.75 | 0.7 | inner_product |
| kron | 20 000 | 0.28 | **0.73** | 0.69 | **0.51** | **0.89** | **0.87** | logn | spmi | 0.75 | 1 | inner_product |
| kron | 30 000 | 0.29 | **0.73** | 0.70 | **0.51** | **0.89** | **0.88** | logn | spmi | 0.75 | 1 | inner_product |
| kron | 40 000 | 0.29 | **0.73** | **0.72** | **0.51** | **0.89** | **0.88** | logn | spmi | 0.75 | 1 | inner_product |
| kron | 50 000 | 0.29 | **0.73** | 0.72 | **0.52** | **0.82** | **0.87** | logn | spmi | 0.75 | 1 | inner_product |

**Table A.10:** Universal (operator-dependent) Heuristics selection



**(a)** SimLex999

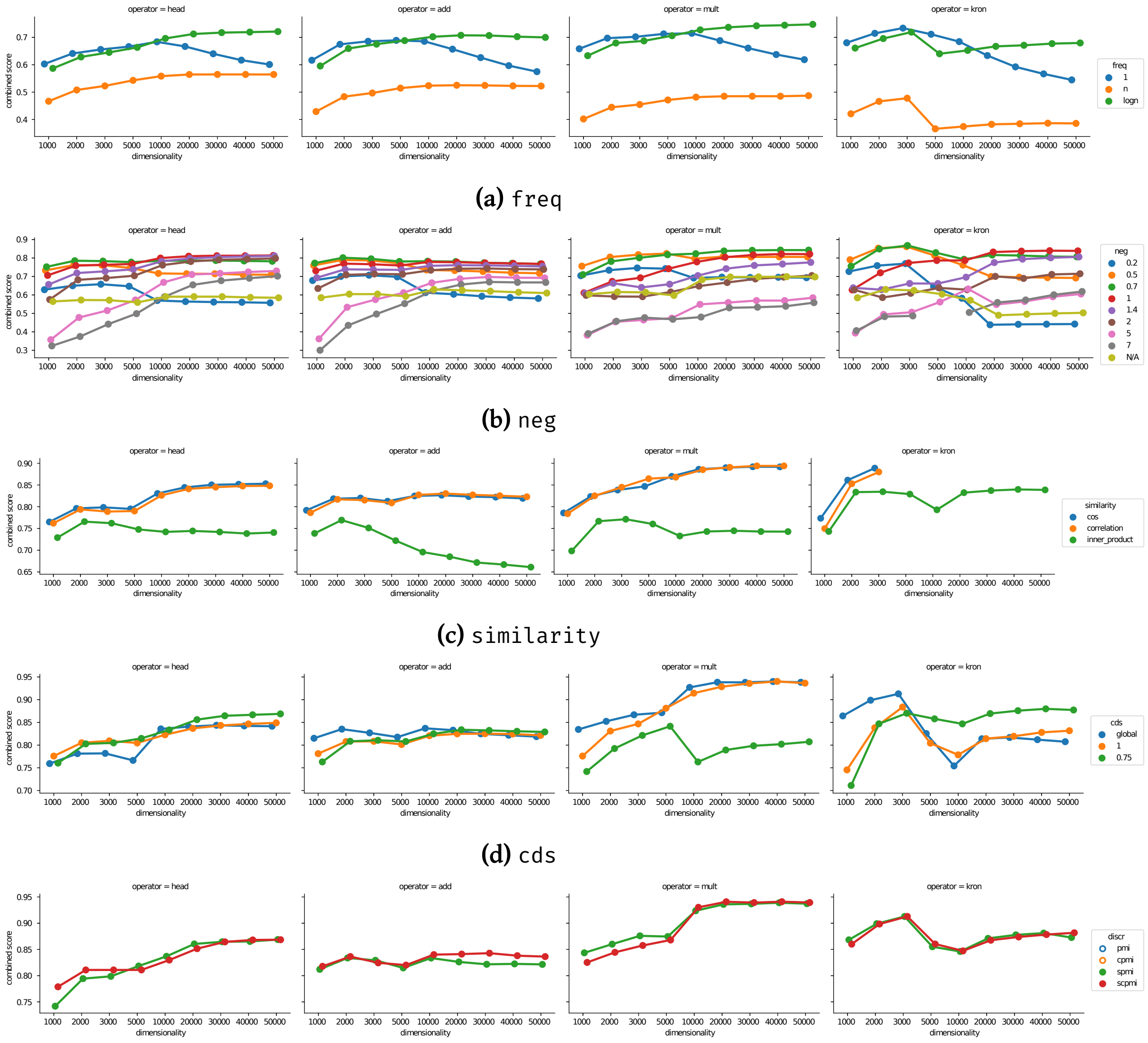
**(b)** MEN

**(c)** KS14

**(d)** GS11

**(e)** PhraseRel

**Figure A.2:** Performance of models based on the selection over the average universal performance



**(a)** `freq`

**(b)** `neg`

**(c)** `similarity`

**(d)** `cds`

**(e)** `discr`

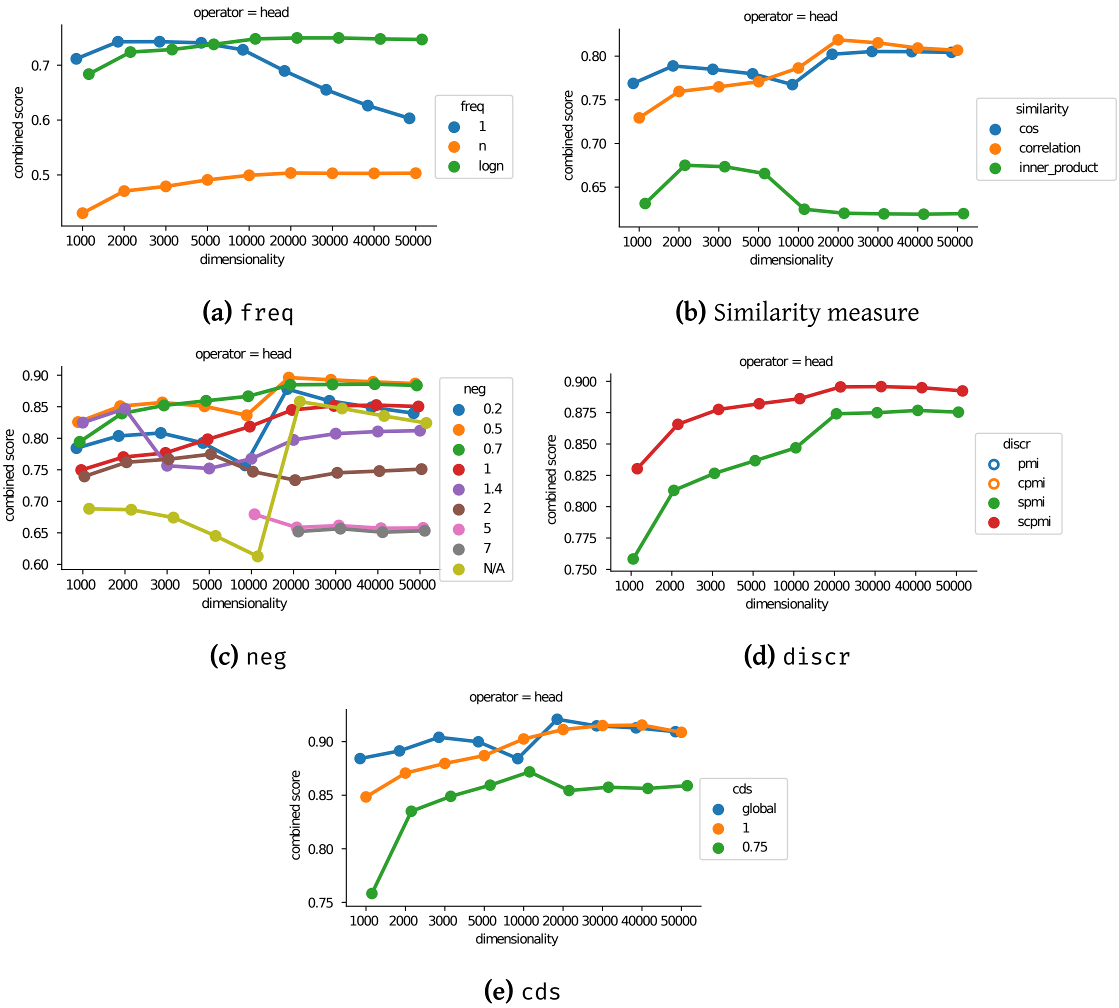
**Figure A.3:** Universal (operator-dependent) parameter influence

| | | | | | | head | | add | | | mult | | | kron | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | SimLex999 | men | KS14 | GS11 | PhraseRel | KS14 | GS11 | PhraseRel | KS14 | GS11 | PhraseRel |
| dimensionality | discr | cds | freq | neg | similarity | | | | | | | | | | | |
| 1 000 | scpmi | global | 1 | 0.7 | cos | 0.33 | 0.65 | **0.74** | **0.32** | **0.86** | **0.73** | 0.44 | **0.89** | **0.76** | **0.43** | **0.86** |
| 2 000 | scpmi | global | 1 | 0.7 | cos | **0.35** | 0.68 | **0.75** | **0.29** | **0.79** | **0.74** | **0.45** | **0.82** | **0.78** | **0.44** | **0.89** |
| 3 000 | scpmi | global | 1 | 0.7 | cos | **0.35** | 0.69 | **0.76** | **0.29** | **0.82** | **0.74** | **0.48** | **0.86** | **0.78** | **0.47** | **0.93** |
| 5 000 | cpmi | 1 | logn | N/A | correlation | **0.35** | 0.73 | **0.78** | 0.25 | **0.82** | **0.75** | 0.43 | **0.89** | | | |
| 10 000 | scpmi | global | logn | 0.7 | correlation | **0.36** | 0.73 | **0.78** | 0.25 | **0.79** | **0.76** | **0.50** | **0.89** | | | |
| 20 000 | cpmi | 1 | logn | N/A | correlation | **0.37** | **0.75** | **0.78** | 0.24 | **0.71** | **0.74** | 0.44 | **1.00** | | | |
| 30 000 | scpmi | 1 | logn | 0.7 | correlation | **0.37** | **0.76** | **0.78** | 0.17 | **0.79** | **0.75** | **0.48** | **0.89** | | | |
| 40 000 | scpmi | 1 | logn | 0.7 | correlation | **0.37** | **0.76** | **0.78** | 0.17 | **0.79** | **0.75** | **0.49** | **0.89** | | | |
| 50 000 | scpmi | global | logn | 0.7 | correlation | **0.37** | **0.75** | **0.76** | 0.20 | **0.71** | **0.74** | **0.50** | **0.89** | | | |

**Table A.11:** Universal (operator-independent) Max selection

|  |  |  |  |  |  | head |  | add |  |  | mult |  |  | kron |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  |  |  |  |  | SimLex999 | men | KS14 | GS11 | PhraseRel | KS14 | GS11 | PhraseRel | KS14 | GS11 | PhraseRel |
| dimensionality | discr | cds | freq | neg | similarity |  |  |  |  |  |  |  |  |  |  |  |
| 1 000 | scpmi | global | 1 | 0.7 | cos | 0.33 | 0.65 | 0.74 | **0.32** | **0.86** | **0.73** | 0.44 | **0.89** | **0.76** | 0.43 | **0.86** |
| 2 000 | scpmi | global | 1 | 0.7 | cos | **0.35** | 0.68 | 0.75 | **0.29** | **0.79** | **0.74** | 0.45 | **0.82** | **0.78** | 0.44 | **0.89** |
| 3 000 | scpmi | global | 1 | 0.7 | cos | **0.35** | 0.69 | 0.76 | **0.29** | **0.82** | **0.74** | 0.48 | **0.86** | **0.78** | **0.47** | **0.93** |
| 5 000 | scpmi | global | 1 | 0.7 | cos | 0.34 | 0.70 | 0.75 | **0.27** | **0.82** | **0.74** | 0.49 | **0.89** |  |  |  |
| 10 000 | scpmi | global | 1 | 0.7 | cos | 0.33 | 0.70 | 0.74 | 0.24 | **0.75** | **0.75** | 0.49 | **0.93** |  |  |  |
| 20 000 | scpmi | global | logn | 0.7 | correlation | **0.37** | **0.75** | **0.77** | 0.23 | **0.75** | **0.75** | **0.50** | **0.89** |  |  |  |
| 30 000 | scpmi | global | logn | 0.7 | correlation | **0.37** | **0.75** | **0.77** | 0.22 | **0.71** | **0.74** | **0.50** | **0.89** |  |  |  |
| 40 000 | scpmi | global | logn | 0.7 | correlation | **0.37** | **0.75** | **0.77** | 0.21 | **0.71** | **0.74** | **0.51** | **0.89** |  |  |  |
| 50 000 | scpmi | global | logn | 0.7 | correlation | **0.37** | **0.75** | 0.76 | 0.20 | **0.71** | **0.74** | **0.50** | **0.89** |  |  |  |

**Table A.12:** Single (operator-independent) heuristics selection



**(a)** `freq`

**(b)** Similarity measure

**(c)** `neg`

**(d)** `discr`

**(e)** `cds`

**Figure A.4:** Universal (parameter independent) parameter influence

| selection | operator | dimensionality | freq | discr | neg | cds | copy-object | copy-subject | frobenius-add | frobenius-mult | frobenius-outer | relational |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| single | | 1 000 | 1 | scpmi | 0.7 | global | **0.60** | **0.70** | **0.71** | **0.69** | **0.73** | **0.73** |
| single | | 2 000 | 1 | scpmi | 0.7 | global | **0.62** | **0.70** | **0.72** | **0.72** | **0.74** | **0.74** |
| single | | 3 000 | 1 | scpmi | 0.7 | global | **0.61** | **0.71** | **0.73** | **0.72** | **0.75** | **0.75** |
| universal (max) | add | 1 000 | 1 | scpmi | 1.0 | global | **0.60** | **0.74** | **0.75** | **0.67** | **0.74** | **0.74** |
| universal (max) | add | 2 000 | 1 | cpmi | N/A | 1 | **0.61** | **0.73** | **0.75** | **0.75** | **0.76** | **0.76** |
| universal (max) | add | 3 000 | logn | cpmi | N/A | 1 | **0.62** | **0.73** | **0.75** | **0.71** | **0.76** | **0.76** |
| universal (max) | mult | 1 000 | logn | spmi | 0.7 | global | **0.60** | **0.71** | **0.72** | **0.68** | **0.73** | **0.74** |
| universal (max) | mult | 2 000 | logn | spmi | 0.2 | 1 | **0.61** | **0.70** | **0.72** | **0.69** | **0.74** | **0.74** |
| universal (max) | mult | 3 000 | logn | spmi | 0.7 | global | **0.62** | **0.72** | **0.74** | **0.71** | **0.74** | **0.74** |
| universal (max) | kron | 1 000 | logn | spmi | 1.0 | global | **0.60** | **0.73** | **0.74** | **0.66** | **0.74** | **0.74** |
| universal (max) | kron | 2 000 | logn | scpmi | 1.0 | global | **0.62** | **0.74** | **0.75** | **0.68** | **0.75** | **0.75** |
| universal (max) | kron | 3 000 | 1 | scpmi | 1.0 | global | **0.63** | **0.73** | **0.76** | **0.69** | **0.77** | **0.77** |
| universal (heuristics) | add | 1 000 | 1 | scpmi | 0.7 | global | **0.59** | **0.72** | **0.73** | **0.73** | **0.74** | **0.74** |
| universal (heuristics) | add | 2 000 | 1 | scpmi | 0.7 | global | **0.61** | **0.72** | **0.74** | **0.73** | **0.75** | **0.75** |
| universal (heuristics) | add | 3 000 | 1 | scpmi | 0.7 | global | **0.61** | **0.72** | **0.75** | **0.73** | **0.75** | **0.75** |
| universal (heuristics) | mult | 1 000 | 1 | spmi | 0.5 | global | **0.59** | **0.71** | **0.72** | **0.72** | **0.73** | **0.73** |
| universal (heuristics) | mult | 2 000 | 1 | spmi | 0.5 | global | **0.60** | **0.72** | **0.73** | **0.73** | **0.74** | **0.74** |
| universal (heuristics) | mult | 3 000 | 1 | spmi | 0.5 | global | **0.61** | **0.71** | **0.74** | **0.73** | **0.74** | **0.74** |
| universal (heuristics) | kron | 1 000 | 1 | spmi | 0.7 | global | **0.60** | **0.71** | **0.72** | **0.69** | **0.73** | **0.74** |
| universal (heuristics) | kron | 2 000 | 1 | spmi | 0.7 | global | **0.62** | **0.72** | **0.74** | **0.71** | **0.74** | **0.75** |
| universal (heuristics) | kron | 3 000 | 1 | spmi | 0.7 | global | **0.62** | **0.71** | **0.74** | **0.72** | **0.74** | **0.75** |

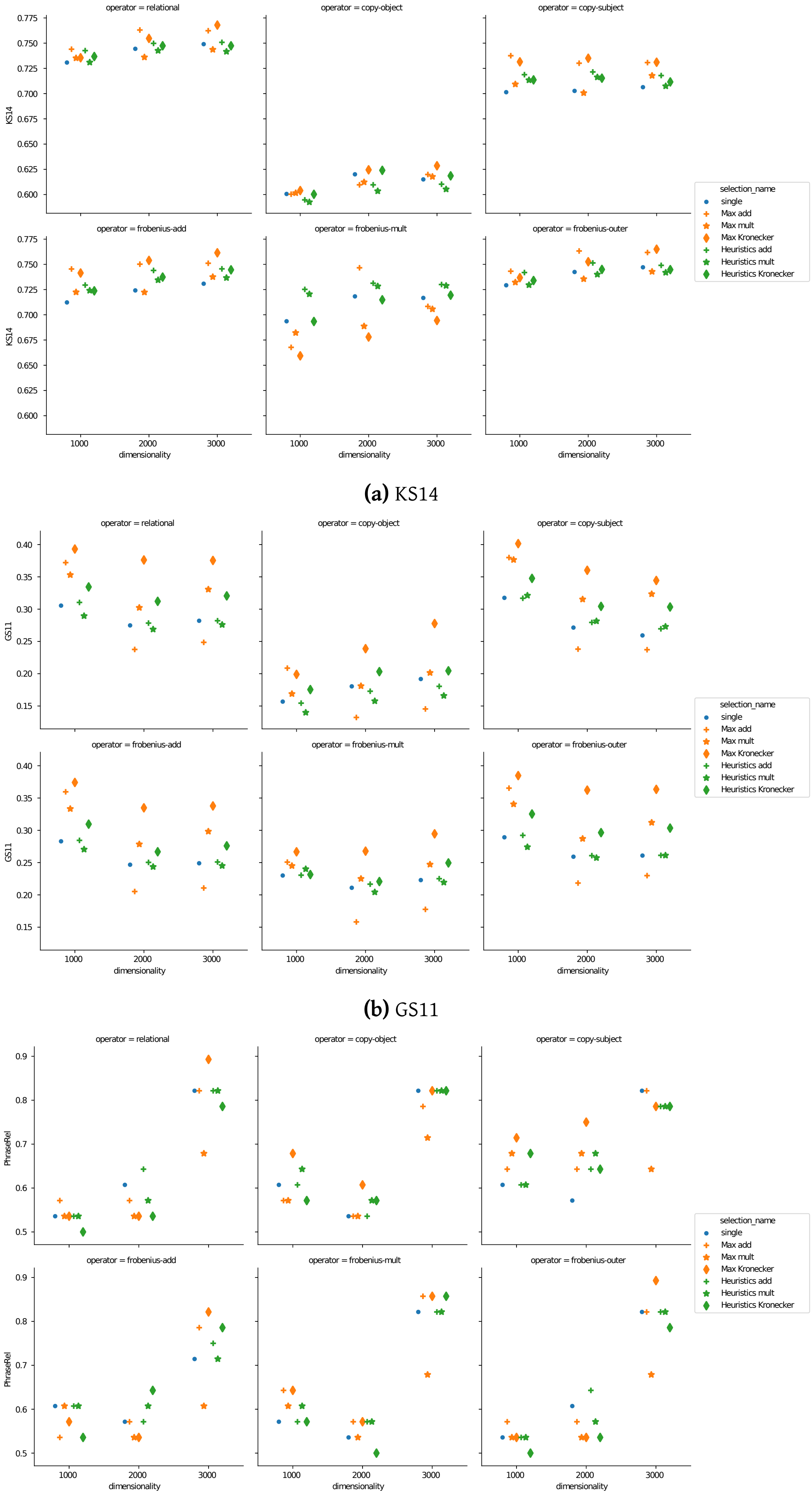
**Table A.13:** Frobenius operators on KS14

| selection | operator | dimensionality | freq | discr | neg | cds | copy-object | copy-subject | frobenius-add | frobenius-mult | frobenius-outer | relational |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| single | | 1 000 | 1 | scpmi | 0.7 | global | 0.16 | 0.32 | 0.28 | 0.23 | 0.29 | 0.31 |
| single | | 2 000 | 1 | scpmi | 0.7 | global | 0.18 | 0.27 | 0.25 | 0.21 | 0.26 | 0.27 |
| single | | 3 000 | 1 | scpmi | 0.7 | global | 0.19 | 0.26 | 0.25 | 0.22 | 0.26 | 0.28 |
| universal (max) | add | 1 000 | 1 | scpmi | 1.0 | global | 0.21 | 0.38 | **0.36** | **0.25** | **0.37** | 0.37 |
| universal (max) | add | 2 000 | 1 | cpmi | N/A | 1 | 0.13 | 0.24 | 0.21 | 0.16 | 0.22 | 0.24 |
| universal (max) | add | 3 000 | logn | cpmi | N/A | 1 | 0.15 | 0.24 | 0.21 | 0.18 | 0.23 | 0.25 |
| universal (max) | mult | 1 000 | logn | spmi | 0.7 | global | 0.17 | **0.38** | 0.33 | 0.25 | **0.34** | 0.35 |
| universal (max) | mult | 2 000 | logn | spmi | 0.2 | 1 | 0.18 | 0.32 | 0.28 | 0.23 | 0.29 | 0.30 |
| universal (max) | mult | 3 000 | logn | spmi | 0.7 | global | 0.20 | 0.32 | 0.30 | **0.25** | 0.31 | 0.33 |
| universal (max) | kron | 1 000 | logn | spmi | N/A | global | 0.20 | **0.40** | **0.37** | **0.27** | **0.38** | **0.39** |
| universal (max) | kron | 2 000 | logn | scpmi | N/A | global | **0.24** | 0.36 | **0.33** | **0.27** | **0.36** | **0.38** |
| universal (max) | kron | 3 000 | 1 | scpmi | N/A | global | **0.28** | 0.34 | **0.34** | **0.29** | **0.36** | **0.38** |
| universal (heuristics) | add | 1 000 | 1 | scpmi | 0.7 | global | 0.15 | 0.32 | 0.28 | 0.23 | 0.29 | 0.31 |
| universal (heuristics) | add | 2 000 | 1 | scpmi | 0.7 | global | 0.17 | 0.28 | 0.25 | 0.22 | 0.26 | 0.28 |
| universal (heuristics) | add | 3 000 | 1 | scpmi | 0.7 | global | 0.18 | 0.27 | 0.25 | 0.23 | 0.26 | 0.28 |
| universal (heuristics) | mult | 1 000 | 1 | spmi | 0.5 | global | 0.14 | 0.32 | 0.27 | **0.24** | 0.27 | 0.29 |
| universal (heuristics) | mult | 2 000 | 1 | spmi | 0.5 | global | 0.16 | 0.28 | 0.24 | 0.20 | 0.26 | 0.27 |
| universal (heuristics) | mult | 3 000 | 1 | spmi | 0.5 | global | 0.17 | 0.27 | 0.25 | 0.22 | 0.26 | 0.28 |
| universal (heuristics) | kron | 1 000 | 1 | spmi | 0.7 | global | 0.18 | **0.35** | 0.31 | 0.23 | **0.33** | 0.33 |
| universal (heuristics) | kron | 2 000 | 1 | spmi | 0.7 | global | 0.20 | 0.30 | 0.27 | 0.22 | 0.30 | 0.31 |
| universal (heuristics) | kron | 3 000 | 1 | spmi | 0.7 | global | 0.20 | 0.30 | 0.28 | **0.25** | 0.30 | 0.32 |

**Table A.14:** Frobenius operators on GS11

| selection | selection_operator | dimensionality | freq | discr | neg | cds | copy-object | copy-subject | frobenius-add | frobenius-mult | frobenius-outer | relational |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| single | | 1 000 | 1 | scpmi | 0.7 | global | **0.61** | **0.61** | **0.61** | **0.57** | **0.54** | **0.54** |
| single | | 2 000 | 1 | scpmi | 0.7 | global | **0.54** | **0.57** | **0.57** | **0.54** | **0.61** | **0.61** |
| single | | 3 000 | 1 | scpmi | 0.7 | global | **0.82** | **0.82** | **0.71** | **0.82** | **0.82** | **0.82** |
| universal (max) | add | 1 000 | 1 | scpmi | 1.0 | global | **0.57** | **0.64** | **0.54** | **0.64** | **0.57** | **0.57** |
| universal (max) | add | 2 000 | 1 | cpmi | N/A | 1 | **0.54** | **0.64** | **0.57** | **0.57** | **0.57** | **0.57** |
| universal (max) | add | 3 000 | logn | cpmi | N/A | 1 | **0.79** | **0.82** | **0.79** | **0.86** | **0.82** | **0.82** |
| universal (max) | mult | 1 000 | logn | spmi | 0.7 | global | **0.57** | **0.68** | **0.61** | **0.61** | **0.54** | **0.54** |
| universal (max) | mult | 2 000 | logn | spmi | 0.2 | 1 | **0.54** | **0.68** | **0.54** | **0.54** | **0.54** | **0.54** |
| universal (max) | mult | 3 000 | logn | spmi | 0.7 | global | **0.71** | **0.64** | **0.61** | **0.68** | **0.68** | **0.68** |
| universal (max) | kron | 1 000 | logn | spmi | 1.0 | global | **0.68** | **0.71** | **0.57** | **0.64** | **0.54** | **0.54** |
| universal (max) | kron | 2 000 | logn | scpmi | 1.0 | global | **0.61** | **0.75** | **0.54** | **0.57** | **0.54** | **0.54** |
| universal (max) | kron | 3 000 | 1 | scpmi | 1.0 | global | **0.82** | **0.79** | **0.82** | **0.86** | **0.89** | **0.89** |
| universal (heuristics) | add | 1 000 | 1 | scpmi | 0.7 | global | **0.61** | **0.61** | **0.61** | **0.57** | **0.54** | **0.54** |
| universal (heuristics) | add | 2 000 | 1 | scpmi | 0.7 | global | **0.54** | **0.64** | **0.57** | **0.57** | **0.64** | **0.64** |
| universal (heuristics) | add | 3 000 | 1 | scpmi | 0.7 | global | **0.82** | **0.79** | **0.75** | **0.82** | **0.82** | **0.82** |
| universal (heuristics) | mult | 1 000 | 1 | spmi | 0.5 | global | **0.64** | **0.61** | **0.61** | **0.61** | **0.54** | **0.54** |
| universal (heuristics) | mult | 2 000 | 1 | spmi | 0.5 | global | **0.57** | **0.68** | **0.61** | **0.57** | **0.57** | **0.57** |
| universal (heuristics) | mult | 3 000 | 1 | spmi | 0.5 | global | **0.82** | **0.79** | **0.71** | **0.82** | **0.82** | **0.82** |
| universal (heuristics) | kron | 1 000 | 1 | spmi | 0.7 | global | **0.57** | **0.68** | **0.54** | **0.57** | **0.50** | **0.50** |
| universal (heuristics) | kron | 2 000 | 1 | spmi | 0.7 | global | **0.57** | **0.64** | **0.64** | **0.50** | **0.54** | **0.54** |
| universal (heuristics) | kron | 3 000 | 1 | spmi | 0.7 | global | **0.82** | **0.79** | **0.79** | **0.86** | **0.79** | **0.79** |

**Table A.15:** Frobenius operators on PhraseRel



**(a)** KS14

**(b)** GS11

**(c)** PhraseRel

**Figure A.5:** Performance of the categorical operators

# Colophon

This thesis was typeset with XƎLaTeX, a TeX typesetting engine that uses Unicode and supports modern font technologies. The main font is Gentium Basic, which is a serif typeface. It supports a wide range of Latin- and Cyrillic-based alphabets. The font is distributed under the SIL Open Font License and available at `http://software.sil.org/gentium/`.

To ease the navigation, the headings are in sans serif typeface **Transport Heavy**, a font designed for British road signs. Transport Heavy is subject to Crown Copyright, and this font contains public sector information licensed under the Open Government Licence v1.0. It was downloaded from `http://www.cbrd.co.uk/fonts/`.

The monospace font is `Fira Mono`, a typeface designed by Mozilla. It is available under the SIL Open Font License, Version 1.1. The typeface is available at `http://mozilla.github.io/Fira/`.

The plots are produced with seaborn, a Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics.