A Systematic Study of Semantic Vector Space Model Parameters

Douwe Kiela
University of Cambridge
Computer Laboratory
douwe.kiela@cl.cam.ac.uk

Stephen Clark
University of Cambridge
Computer Laboratory
sc609@cam.ac.uk

Abstract

We present a systematic study of parameters used in the construction of semantic vector space models. Evaluation is carried out on a variety of similarity tasks, including a compositionality dataset, using several source corpora. In addition to recommendations for optimal parameters, we present some novel findings, including a similarity metric that outperforms the alternatives on all tasks considered.

1 Introduction

Vector space models (VSMs) represent the meanings of lexical items as vectors in a "semantic space". The benefit of VSMs is that they can easily be manipulated using linear algebra, allowing a degree of similarity between vectors to be computed. They rely on the *distributional hypothesis* (Harris, 1954): the idea that "words that occur in similar contexts tend to have similar meanings" (Turney and Pantel, 2010; Erk, 2012). The construction of a suitable VSM for a particular task is highly parameterised, and there appears to be little consensus over which parameter settings to use.

This paper presents a systematic study of the following parameters:

- vector size;
- window size;
- window-based or dependency-based context;
- feature granularity;
- similarity metric;
- weighting scheme;
- stopwords and high frequency cut-off.

A representative set of semantic similarity datasets has been selected from the literature, including a phrasal similarity dataset for evaluating compositionality. The choice of source corpus is likely to influence the quality of the VSM, and so we use a selection of source corpora. Hence there are two additional "superparameters":

- dataset for evaluation;
- source corpus.

Previous studies have been limited to investigating only a small number of parameters, and using a limited set of source corpora and tasks for evaluation (Curran and Moens, 2002a; Curran and Moens, 2002b; Curran, 2004; Grefenstette, 1994; Pado and Lapata, 2007; Sahlgren, 2006; Turney and Pantel, 2010; Schulte im Walde et al., 2013). Rohde et al. (2006) considered several weighting schemes for a large variety of tasks, while Weeds et al. (2004) did the same for similarity metrics. Stone et al. (2008) investigated the effectiveness of sub-spacing corpora, where a larger corpus is queried in order to construct a smaller sub-spaced corpus (Zelikovitz and Kogan, 2006). Blacoe and Lapata (2012) compare several types of vector representations for semantic composition tasks. The most comprehensive existing studies of VSM parameters — encompassing window sizes, feature granularity, stopwords and dimensionality reduction — are by Bullinaria and Levy (2007; 2012) and Lapesa and Evert (2013).

Section 2 introduces the various parameters of vector space model construction. We then attempt, in Section 3, to answer some of the fundamental questions for building VSMs through a number of experiments that consider each of the selected parameters. In Section 4 we examine how these findings relate to the recent development of distributional compositional semantics (Baroni et al., 2013; Clark, 2014), where vectors for words are combined into vectors for phrases.

2 Data and Parameters

Two datasets have dominated the literature with respect to VSM parameters: WordSim353 (Finkelstein et al., 2002) and the TOEFL synonym dataset

Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC) @ EACL 2014, pages 21–30,
Gothenburg, Sweden, April 26-30 2014.

Dataset	Pairings	Words
RG	65	48
MC	30	39
W353	353	437
MEN	3000	751
TOEFL	80	400
M&L10	324	314

Table 1: Datasets for evaluation

(Landauer and Dumais, 1997). There is a risk that semantic similarity studies have been overfitting to their idiosyncracies, so in this study we evaluate on a variety of datasets: in addition to WordSim353 (W353) and TOEFL, we also use the Rubenstein & Goodenough (RG) (1965) and Miller & Charles (MC) (1991) data, as well as a much larger set of similarity ratings: the MEN dataset (Bruni et al., 2012). All these datasets consist of human similarity ratings for word pairings, except TOEFL, which consists of multiple choice questions where the task is to select the correct synonym for a target word. In Section 4 we examine our parameters in the context of distributional compositional semantics, using the evaluation dataset from Mitchell and Lapata (2010). Table 1 gives statistics for the number of words and word pairings in each of the datasets.

As well as using a variety of datasets, we also consider three different corpora from which to build the vectors, varying in size and domain. These include the **BNC** (Burnard, 2007) (10^6 word types, 10^8 tokens) and the larger **ukWaC** (Baroni et al., 2009) (10^7 types, 10^9 tokens). We also include a **sub-spaced Wikipedia** corpus (Stone et al., 2008): for all words in the evaluation datasets, we build a subcorpus by querying the top 10-ranked Wikipedia documents using the words as search terms, resulting in a corpus with 10^6 word types and 10^7 tokens. For examining the dependency-based contexts, we include the **Google Syntactic N-gram** corpus (Goldberg and Orwant, 2013), with 10^7 types and 10^{11} tokens.

2.1 Parameters

We selected the following set of parameters for investigation, all of which are fundamental to vector space model construction[1].

[1]Another obvious parameter would be dimensionality reduction, which we chose not to include because it does not represent a fundamental aspect of VSM construction: dimensionality reduction relies on some original non-reduced model, and directly depends on its quality.

Vector size Each component of a vector represents a context (or perhaps more accurately a "contextual element", such as second word to the left of the target word).[2] The number of components varies hugely in the literature, but a typical value is in the low thousands. Here we consider vector sizes ranging from 50,000 to 500,000, to see whether larger vectors lead to better performance.

Context There are two main approaches to modelling context: window-based and dependency-based. For window-based methods, contexts are determined by word co-occurrences within a window of a given size, where the window simply spans a number of words occurring around instances of a target word. For dependency-based methods, the contexts are determined by word co-occurrences in a particular syntactic relation with a target word (e.g. target word *dog* is the subject of *run*, where *run*_subj is the context). We consider different window sizes and compare window-based and dependency-based methods.

Feature granularity Context words, or "features", are often stemmed or lemmatised. We investigate the effect of stemming and lemmatisation, in particular to see whether the effect varies with corpus size. We also consider more fine-grained features in which each context word is paired with a POS tag or a lexical category from CCG (Steedman, 2000).

Similarity metric A variety of metrics can be used to calculate the similarity between two vectors. We consider the similarity metrics in Table 2.

Weighting Weighting schemes increase the importance of contexts that are more indicative of the meaning of the target word: the fact that *cat* co-occurs with *purr* is much more informative than its co-occurrence with *the*. Table 3 gives definitions of the weighting schemes considered.

Stopwords, high frequency cut-off Function words and stopwords are often considered too uninformative to be suitable context words. Ignoring them not only leads to a reduction in model size and computational effort, but also to a more informative distributional vector. Hence we followed standard practice and did not use stopwords as context words (using the stoplist in NLTK (Bird et al., 2009)). The question we investigated is

[2]We will use the term "feature" or "context" or "context word" to refer to contextual elements.

Measure	Definition
Euclidean	$\frac{1}{1+\sqrt{\sum_{i=1}^{n}(u_i-v_i)^2}}$
Cityblock	$\frac{1}{1+\sum_{i=1}^{n}\lvert u_i-v_i\rvert}$
Chebyshev	$\frac{1}{1+\max_i\lvert u_i-v_i\rvert}$
Cosine	$\frac{u\cdot v}{\lvert u\rvert\lvert v\rvert}$
Correlation	$\frac{(u-\mu_u)\cdot(v-\mu_v)}{\lvert u\rvert\lvert v\rvert}$
Dice	$\frac{2\sum_{i=0}^{n} min(u_i,v_i)}{\sum_{i=0}^{n} u_i+v_i}$
Jaccard	$\frac{u\cdot v}{\sum_{i=0}^{n} u_i+v_i}$
Jaccard2	$\frac{\sum_{i=0}^{n} min(u_i,v_i)}{\sum_{i=0}^{n} max(u_i,v_i)}$
Lin	$\frac{\sum_{i=0}^{n} u_i+v_i}{\lvert u\rvert+\lvert v\rvert}$
Tanimoto	$\frac{u\cdot v}{\lvert u\rvert+\lvert v\rvert-u\cdot v}$
Jensen-Shannon Div	$1-\frac{\frac{1}{2}(D(u\lvert\lvert\frac{u+v}{2})+D(v\lvert\lvert\frac{u+v}{2}))}{\sqrt{2\log 2}}$
α-skew	$1-\frac{D(u\lvert\lvert\alpha v+(1-\alpha)u)}{\sqrt{2\log 2}}$

Table 2: Similarity measures between vectors v and u, where v_i is the ith component of v

whether removing more context words, based on a frequency cut-off, can improve performance.

3 Experiments

The parameter space is too large to analyse exhaustively, and so we adopted a strategy for how to navigate through it, selecting certain parameters to investigate first, which then get fixed or "clamped" in the remaining experiments. Unless specified otherwise, vectors are generated with the following restrictions and transformations on features: stopwords are removed, numbers mapped to 'NUM', and only strings consisting of alphanumeric characters are allowed. In all experiments, the features consist of the frequency-ranked first n words in the given source corpus.

Four of the five similarity datasets (RG, MC, W353, MEN) contain continuous scales of similarity ratings for word pairs; hence we follow standard practice in using a Spearman correlation coefficient ρ_s for evaluation. The fifth dataset (TOEFL) is a set of multiple-choice questions, for which an accuracy measure is appropriate. Calculating an aggregate score over all datasets is non-trivial, since taking the mean of correlation scores leads to an under-estimation of performance; hence for the aggregate score we use the Fisher-transformed z-variable of the correlation datasets, and take the weighted average of its inverse over the correlation datasets and the TOEFL accuracy score (Silver and Dunlap, 1987).

Scheme	Definition
None	$w_{ij}=f_{ij}$
TF-IDF	$w_{ij}=\log(f_{ij})\times\log(\frac{N}{n_j})$
TF-ICF	$w_{ij}=\log(f_{ij})\times\log(\frac{N}{f_j})$
Okapi BM25	$w_{ij}=\frac{f_{ij}}{0.5+1.5\times\frac{f_j}{\frac{f_j}{j}}+f_{ij}}\log\frac{N-n_j+0.5}{f_{ij}+0.5}$
ATC	$w_{ij}=\frac{(0.5+0.5\times\frac{f_{ij}}{max_f})\log(\frac{N}{n_j})}{\sqrt{\sum_{i=1}^{N}[(0.5+0.5\times\frac{f_{ij}}{max_f})\log(\frac{N}{n_j})]^2}}$
LTU	$w_{ij}=\frac{(\log(f_{ij})+1.0)\log(\frac{N}{n_j})}{0.8+0.2\times f_j\times\frac{j}{f_j}}$
MI	$w_{ij}=\log\frac{P(t_{ij}\lvert c_j)}{P(t_{ij})P(c_j)}$
PosMI	$\max(0,\text{MI})$
T-Test	$w_{ij}=\frac{P(t_{ij}\lvert c_j)-P(t_{ij})P(c_j)}{\sqrt{P(t_{ij})P(c_j)}}$
χ^2	see (Curran, 2004, p. 83)
Lin98a	$w_{ij}=\frac{f_{ij}\times f}{f_i\times f_j}$
Lin98b	$w_{ij}=-1\times\log\frac{n_j}{N}$
Gref94	$w_{ij}=\frac{\log f_{ij}+1}{\log n_j+1}$

Table 3: Term weighting schemes. f_{ij} denotes the target word frequency in a particular context, f_i the total target word frequency, f_j the total context frequency, N the total of all frequencies, n_j the number of non-zero contexts. $P(t_{ij}|c_j)$ is defined as $\frac{f_{ij}}{f_j}$ and $P(t_{ij})$ as $\frac{f_{ij}}{N}$.

3.1 Vector size

The first parameter we investigate is vector size, measured by the number of features. Vectors are constructed from the BNC using a window-based method, with a window size of 5 (2 words either side of the target word). We experiment with vector sizes up to 0.5M features, which is close to the total number of context words present in the entire BNC according to our preprocessing scheme. Features are added according to frequency in the BNC, with increasingly more rare features being added. For weighting we consider both Positive Mutual Information and T-Test, which have been found to work best in previous research (Bullinaria and Levy, 2012; Curran, 2004). Similarity is computed using Cosine.

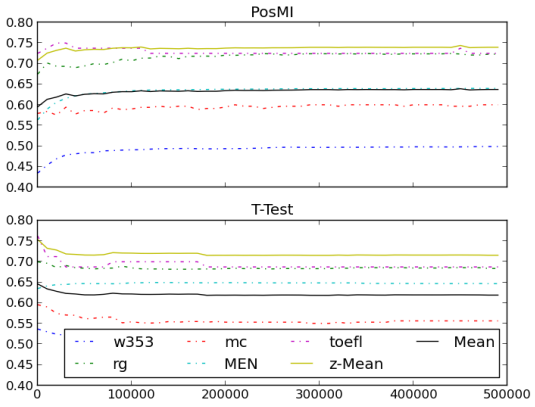

Figure 1: Impact of vector size on performance across different datasets

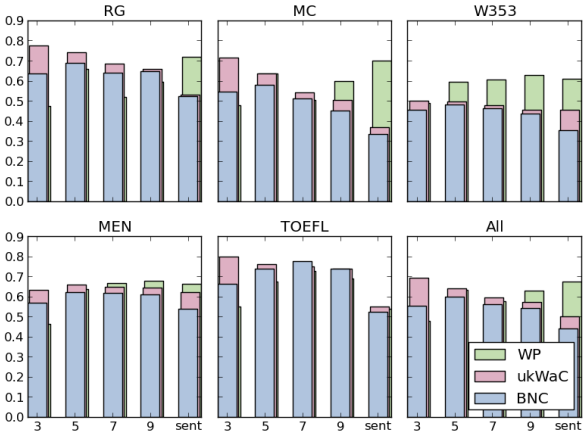

Figure 2: Impact of window size across three corpora

The results in Figure 1 show a clear trend: for both weighting schemes, performance no longer improves after around 50,000 features; in fact, for T-test weighting, and some of the datasets, performance initially declines with an increase in features. Hence we conclude that continuing to add more rare features is detrimental to performance, and that 50,000 features or less will give good performance. An added benefit of smaller vectors is the reduction in computational cost.

3.2 Window size

Recent studies have found that the best window size depends on the task at hand. For example, Hill et al. (2013) found that smaller windows work best for measuring similarity of concrete nouns, whereas larger window sizes work better for abstract nouns. Schulte im Walde et al. (2013) found that a large window size worked best for a compositionality dataset of German noun-noun compounds. Similar relations between window size and performance have been found for similar versus related words, as well as for similar versus associated words (Turney and Pantel, 2010).

We experiment with window sizes of 3, 5, 7, 9 and a full sentence. (A window size of n implies $\frac{n-1}{2}$ words either side of the target word.) We use Positive Mutual Information weighting, Cosine similarity, and vectors of size 50,000 (based on the results from Section 3.1). Figure 2 shows the results for all the similarity datasets, with the aggregated score at the bottom right.

Performance was evaluated on three corpora, in order to answer three questions: Does window size affect performance? Does corpus size interact with window size? Does corpus subspacing interact with window size? Figure 2 clearly shows the answer to all three questions is "yes". First, ukWaC consistently outperforms the BNC, across all window sizes, indicating that a larger source corpus leads to better performance. Second, we see that the larger ukWaC performs better with smaller window sizes compared to the BNC, with the best ukWaC performance typically being found with a window size of only 3. For the BNC, it appears that a larger window is able to offset the smaller size of corpus to some extent.

We also evaluated on a sub-spaced Wikipedia source corpus similar to Stone et al. (2008), which performs much better with larger window sizes than the BNC or ukWaC. Our explanation for this result is that sub-spacing, resulting from searching for Wikipedia pages with the appropriate target terms, provides a focused, less noisy corpus in which context words some distance from the target word are still relevant to its meaning.

In summary, the highest score is typically achieved with the largest source corpora and smallest window size, with the exception of the much smaller sub-spaced Wikipedia corpus.

3.3 Context

The notion of context plays a key role in VSMs. Pado and Lapata (2007) present a comparison of window-based versus dependency-based methods and conclude that dependency-based contexts give better results. We also compare window-based and dependency-based models.

Dependency-parsed versions of the BNC and ukWaC were used to construct syntactically-informed vectors, with a single, labelled arc be-

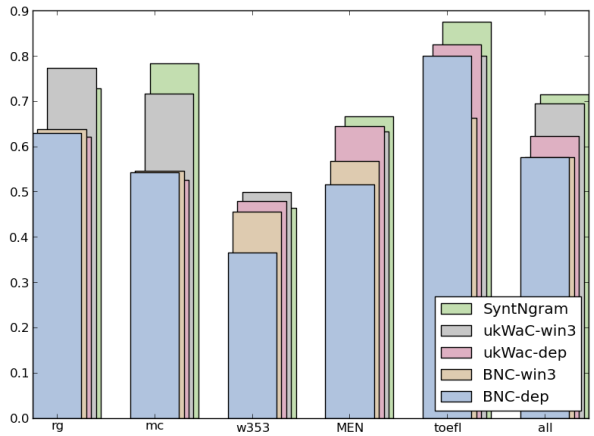

Figure 3: Window versus dependency contexts

tween the target word and context word.[3] Since this effectively provides a window size of 3, we also use a window size of 3 for the window-based method (which provided the best results in Section 3.2 with the ukWaC corpus). As well as the ukWaC and BNC source corpora, we use the Google syntactic N-gram corpus (Goldberg and Orwant, 2013), which is one of the largest corpora to date, and which consists of syntactic n-grams as opposed to window-based n-grams. We use vectors of size 50,000 with Positive Mutual Information weighting and Cosine similarity. Due to its size and associated computational cost, we used only 10,000 contexts for the vectors generated from the syntactic N-gram corpus. The results are shown in Figure 3.

In contrast to the idea that dependency-based methods outperform window-based methods, we find that the window-based models outperform dependency-based models when they are constructed from the same corpus using the small window size. However, Google's syntactic N-gram corpus does indeed outperform window-based methods, even though smaller vectors were used for the Google models (10,000 vs. 50,000 features). We observe large variations across datasets, with window-based methods performing particularly well on some, but not all. In particular, window-based methods clearly outperform dependency-based methods on the RG dataset (for the same source corpus), whereas the opposite trend is observed for the TOEFL synonym dataset. The summary is that the model built from the syntactic N-grams is the overall winner, but when we compare both methods on the same corpus, the window-based method on a large corpus appears to work best (given the small window size).

3.4 Feature granularity

Stemming and lemmatisation are standard techniques in NLP and IR to reduce data sparsity. However, with large enough corpora it may be that the loss of information through generalisation hurts performance. In fact, it may be that increased granularity – through the use of grammatical tags – can lead to improved performance. We test these hypotheses by comparing four types of processed context words: lemmatised, stemmed, POS-tagged, and tagged with CCG lexical categories (which can be thought of as fine-grained POS tags (Clark and Curran, 2007)).[4] The source corpora are BNC and ukWaC, using a window-based method with windows of size 5, Positive Mutual Information weighting, vectors of size 50,000 and Cosine similarity. The results are reported in Figure 4.

The ukWaC-generated vectors outperform the BNC-generated ones on all but a single instance for each of the granularities. Stemming yields the best overall performance, and increasing the granularity does not lead to better results. Even with a very large corpus like ukWaC, stemming yields signficantly better results than not reducing the feature granularity at all. Conversely, apart from the results on the TOEFL synonym dataset, increasing the feature granularity of contexts by including POS tags or CCG categories does not yield any improvement.

3.5 Similarity-weighting combination

There is contrasting evidence in the literature regarding which combination of similarity metric and weighting scheme works best. Here we investigate this question using vectors of size 50,000, no processing of the context features (i.e., "normal" feature granularity), and a window-based method with a window size of 5. Aggregated scores across the datasets are reported in Tables 4 and 5 for the BNC and ukWaC, respectively.

There are some clear messages to be taken from these large tables of results. First, two weighting schemes perform better than the others: Positive Mutual Information (PosMI) and T-Test. On the BNC, the former yields the best results. There are

[3] The Clark and Curran (2007) parser was used to provide the dependencies.

[4] Using NLTK's Porter stemmer and WordNet lemmatiser.

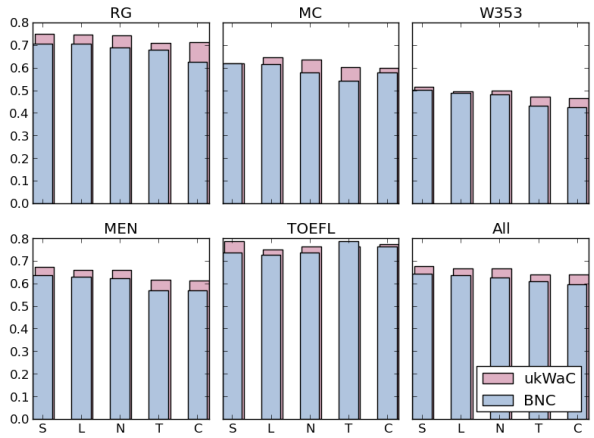

Figure 4: Feature granularity: stemmed (S), lemmatised (L), normal (N), POS-tagged (T) and CCG-tagged (C)

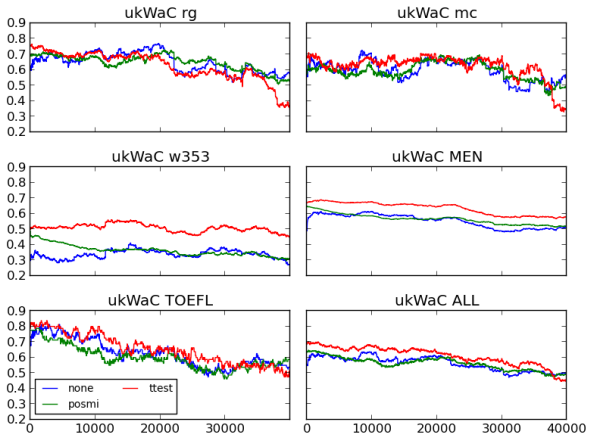

Figure 5: Finding the optimal "contiguous subvector" of size 10,000

	RG	MC	W353	MEN	TOEFL
P+COS	0.74	0.64	0.50	0.66	0.76
P+COR	0.74	0.65	**0.58**	**0.71**	**0.83**
T+COS	0.78	0.69	0.54	0.68	0.78
T+COR	**0.78**	**0.71**	0.54	0.68	0.78

Table 6: Similarity scores on individual datasets for positive mutual information (P) and T-test (T) weighting, with cosine (COS) and correlation (COR) similarity

three similarity metrics that perform particularly well: Cosine, Correlation and the Tanimoto coefficient (the latter also being similar to Cosine; see Table 2). The Correlation similarity metric has the most consistent performance across the different weighting schemes, and yields the highest score for both corpora. The most consistent weighting scheme across the two source corpora and similarity metrics appears to be PosMI.

The highest combined aggregate score is that of PosMI with the Correlation metric, in line with the conclusion of Bullinaria and Levy (2012) that PosMI is the best weighting scheme[5]. However, for the large ukWaC corpus, T-Test achieves similarly high aggregate scores, in line with the work of Curran (2004). When we look at these two weighting schemes in more detail, we see that T-Test works best for the RG and MC datasets, while PosMI works best for the others; see Table 6. Correlation is the best similarity metric in all cases.

[5]In some cases, the combination of weighting scheme and similarity metric results in a division by zero or leads to taking the logarithm of a negative number, in which cases we report the aggregate scores as nan (not-a-number).

3.6 Optimal subvector

Stopwords are typically removed from vectors and not used as features. However, Bullinaria and Levy (2012) find that removing stopwords has no effect on performance. A possible explanation is that, since they are using a weighting scheme, the weights of stopwords are low enough that they have effectively been removed anyhow. This raises the question: are we removing stopwords because they contribute little towards the meaning of the target word, or are we removing them because they have high frequency?

The experiment used ukWaC, with a window-based method and window size of 5, normal feature granularity, Cosine similarity and a sliding vector of size 10,000. Having a sliding vector implies that we throw away up to the first 40,000 contexts as we slide across to the 50,000 mark (replacing the higher frequency contexts with lower frequency ones). In effect, we are trying to find the cut-off point where the 10,000-component "contiguous subvector" of the target word vector is optimal (where the features are ordered by frequency). Results are given for PosMI, T-Test and no weighting at all.

The results are shown in Figure 5. T-test outperforms PosMI at the higher frequency ranges (to the left of the plots) but PosMI gives better results for some of the datasets further to the right. For both weighting schemes the performance decreases as high frequency contexts are replaced with lower frequency contexts.

A different picture emerges when no weighting is used, however. Here the performance can *increase* as high-frequency contexts are replaced

British National Corpus												
	COS	COR	DIC	JC1	JC2	TAN	LIN	EUC	CIB	CHS	JSD	ASK
none	0.49	0.50	0.34	0.35	0.27	0.22	0.30	0.09	0.11	0.08	0.45	0.36
tfidf	0.43	0.44	0.33	0.34	0.22	0.16	0.27	0.13	0.12	0.16	0.38	0.32
tficf	0.47	0.48	0.34	0.36	0.23	0.16	0.27	0.13	0.12	0.15	0.40	0.33
okapi	0.40	0.42	0.37	0.42	0.22	0.23	0.26	0.25	0.15	0.14	0.37	0.26
atc	0.40	0.43	0.25	0.24	0.16	0.34	0.30	0.10	0.13	0.08	0.33	0.23
ltu	0.44	0.45	0.35	0.36	0.22	0.23	0.26	0.22	0.13	0.21	0.37	0.27
mi	0.58	0.61	0.31	0.56	0.29	-0.07	0.45	0.15	0.10	0.09	0.16	-0.04
posmi	0.63	0.66	0.52	0.58	0.35	-0.08	0.45	0.15	0.11	0.06	0.54	0.46
ttest	0.63	0.62	0.11	0.34	0.08	0.63	0.17	0.18	0.14	0.11	nan	nan
chisquared	0.50	0.50	0.46	0.42	0.42	0.42	nan	0.06	0.07	0.08	0.57	0.52
lin98b	0.47	0.52	0.35	0.40	0.21	-0.10	0.29	0.10	0.11	nan	0.38	0.29
gref94	0.46	0.49	0.35	0.37	0.23	0.06	0.28	0.12	0.11	0.09	0.41	0.30

Table 4: Aggregated scores for combinations of weighting schemes and similarity metrics using the BNC. The similarity metrics are Cosine (COS), Correlation (COR), Dice (DIC), Jaccard (JC1), Jaccard2 (JC2), Tanimoto (TAN), Lin (LIN), Euclidean (EUC), CityBlock (CIB), Chebyshev (CHS), Jensen-Shannon Divergence (JSD) and α-skew (ASK)

ukWaC												
	COS	COR	DIC	JC1	JC2	TAN	LIN	EUC	CIB	CHS	JSD	ASK
none	0.55	0.55	0.28	0.35	0.24	0.41	0.31	0.06	0.09	0.08	0.56	0.49
tfidf	0.45	0.47	0.26	0.30	0.20	0.28	0.22	0.14	0.12	0.16	0.37	0.27
tficf	0.45	0.49	0.27	0.33	0.20	0.29	0.24	0.13	0.11	0.09	0.37	0.28
okapi	0.37	0.42	0.33	0.37	0.18	0.27	0.26	0.26	0.17	0.12	0.34	0.20
atc	0.34	0.42	0.13	0.13	0.08	0.15	0.28	0.10	0.09	0.07	0.28	0.15
ltu	0.43	0.48	0.30	0.34	0.19	0.26	0.25	0.26	0.16	0.24	0.36	0.23
mi	0.51	0.53	0.18	0.51	0.16	0.28	0.37	0.18	0.10	0.09	0.12	nan
posmi	0.67	0.70	0.56	0.62	0.42	0.59	0.52	0.23	0.15	0.06	0.60	0.49
ttest	0.70	0.70	0.16	0.48	0.10	0.70	0.22	0.16	0.11	0.15	nan	nan
chisquared	0.57	0.58	0.52	0.56	0.44	0.52	nan	0.08	0.06	0.10	0.63	0.60
lin98b	0.43	0.63	0.31	0.37	0.20	0.23	0.26	0.09	0.10	nan	0.34	0.24
gref94	0.48	0.54	0.27	0.33	0.20	0.17	0.23	0.13	0.11	0.09	0.38	0.25

Table 5: Aggregated scores for combinations of weighting schemes and similarity metrics using ukWaC

with lower-frequency ones, with optimal performance comparable to when weighting is used. There are some scenarios where it may be advantageous not to use weighting, for example in an online setting where the total set of vectors is not fixed; in situations where use of a dimensionality reduction technique does not directly allow for weighting, such as random indexing (Sahlgren, 2006); as well as in settings where calculating weights is too expensive. Where to stop the sliding window varies with the datasets, however, and so our conclusion is that the default scheme should be weighting plus high frequency contexts.

4 Compositionality

In order to examine whether optimal parameters carry over to vectors that are combined into phrasal vectors using a composition operator, we perform a subset of our experiments on the canonical compositionality dataset from Mitchell and Lapata (2010), using vector addition and pointwise multiplication (the best performing operators in the original study).

We evaluate using two source corpora (the BNC and ukWaC) and two window sizes (small, with a window size of 3; and big, where the full sentence is the window). In addition to the weighting schemes from the previous experiment, we include Mitchell & Lapata's own weighting scheme, which (in our notation) is defined as $w_{ij} = \frac{f_{ij} \times N}{f_i \times f_j}$. While all weighting schemes and similarity metrics were tested, we report only the best performing ones: correlations below 0.5 were ommitted for the sake of brevity. Table 7 shows the results.

We find that many of our findings continue to hold. PosMI and T-Test are the best performing weighting schemes, together with Mitchell & Lapata's own weighting scheme. We find that addition outperforms multiplication (contrary to the original study) and that small window sizes work best, except in the VO case. Performance across corpora is comparable. The best performing similarity metrics are Cosine and Correlation, with the latter having a slight edge over the former.

BNC - Small window				
	AN	NN	VO	ALL
add-posmi-cosine	0.57	0.56	0.52	0.55
add-posmi-correlation	**0.66**	**0.60**	0.53	**0.60**
add-ttest-cosine	0.59	0.54	0.53	0.56
add-ttest-correlation	0.60	0.54	0.53	0.56
add-mila-correlation	0.64	0.38	0.51	0.51
ukWaC - Small window				
	AN	NN	VO	ALL
add-posmi-correlation	0.64	0.59	0.56	0.59
add-ttest-cosine	0.61	0.55	0.53	0.56
add-ttest-correlation	0.61	0.55	0.53	0.56
add-mila-correlation	0.64	0.48	0.57	0.56
mult-mila-correlation	0.52	0.44	0.63	0.53
BNC - Large window				
	AN	NN	VO	ALL
add-posmi-correlation	0.47	0.49	0.57	0.51
add-ttest-cosine	0.50	0.53	0.60	0.54
add-ttest-correlation	0.50	0.53	0.60	0.54
add-mila-correlation	0.51	0.49	0.61	0.54
mult-posmi-correlation	0.48	0.48	0.66	0.54
mult-mila-correlation	0.53	0.51	0.67	0.57
ukWaC - Large window				
	AN	NN	VO	ALL
add-posmi-correlation	0.46	0.44	0.60	0.50
add-ttest-cosine	0.46	0.46	0.59	0.50
add-ttest-correlation	0.47	0.46	0.60	0.51
add-mila-correlation	0.47	0.46	0.64	0.52
mult-posmi-correlation	0.44	0.46	0.65	0.52
mult-mila-correlation	0.56	0.49	**0.70**	0.58

Table 7: Selected Spearman ρ scores on the Mitchell & Lapata 2010 compositionality dataset

5 Conclusion

Our experiments were designed to investigate a wide range of VSM parameters, using a variety of evaluation tasks and several source corpora. Across each of the experiments, results are competitive with the state of the art. Some important messages can be taken away from this study:

Experiment 1 Larger vectors do not always lead to better performance. As vector size increases, performance stabilises, and a vector size of around 50,000 appears to be optimal.

Experiment 2 The size of the window has a clear impact on performance: a large corpus with a small window size performs best, but high performance can be achieved on a small subspaced corpus, if the window size is large.

Experiment 3 The size of the source corpus is more important than whether the model is window- or dependency-based. Window-based methods with a window size of 3 yield better results than dependency-based methods with a window of 3 (i.e. having a single arc). The Google Syntactic N-gram corpus yields very good performance, but it is unclear whether this is due to being dependency-based or being very large.

Experiment 4 The granularity of the context words has a relatively low impact on performance, but stemming yields the best results.

Experiment 5 The optimal combination of weighting scheme and similarity metric is Positive Mutual Information with a mean-adjusted version of Cosine that we have called Correlation. Another high-performing weighting scheme is T-Test, which works better for smaller vector sizes. The Correlation similarity metric consistently outperforms Cosine, and we recommend its use.

Experiment 6 Use of a weighting scheme obviates the need for removing high-frequency features. Without weighting, many of the high-frequency features should be removed. However, if weighting is an option we recommend its use.

Compositionality The best parameters for individual vectors generally carry over to a compositional similarity task where phrasal similarity is evaluated by combining vectors into phrasal vectors.

Furthermore, we observe that in general performance increases as source corpus size increases, so we recommend using a corpus such as ukWaC over smaller corpora like the BNC. Likewise, since the MEN dataset is the largest similarity dataset available and mirrors our aggregate score the best across the various experiments, we recommend evaluating on that similarity task if only a single dataset is used for evaluation.

Obvious extensions include an analysis of the performance of the various dimensionality reduction techniques, examining the importance of window size and feature granularity for dependency-based methods, and further exploring the relation between the size and frequency distribution of a corpus together with the optimal characteristics (such as the high-frequency cut-off point) of vectors generated from that source.

Acknowledgments

This work has been supported by EPSRC grant EP/I037512/1. We would like to thank Laura Rimell, Tamara Polajnar and Felix Hill for helpful comments and suggestions.

References

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

Marco Baroni, Raffaella Bernardi, and Roberto Zamparelli. 2013. Frege in Space: A program for compositional distributional semantics. *Linguistic Issues in Language Technologies (LiLT)*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.

William Blacoe and Mirella Lapata. 2012. 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*, pages 546–556, Jeju Island, Korea, July. Association for Computational Linguistics.

Elia Bruni, Gemma Boleda, Marco Baroni, and N. K. Tran. 2012. Distributional Semantics in Technicolor. In *Proceedings of the ACL 2012*.

John A. Bullinaria and Joseph P. Levy. 2007. Extracting Semantic Representations from Word Co-occurrence Statistics: A computational study. *Behavior Research Methods*, 39:510–526.

John A. Bullinaria and Joseph P. Levy. 2012. Extracting Semantic Representations from Word Co-occurrence Statistics: Stop-lists, Stemming and SVD. *Behavior Research Methods*, 44:890–907.

L. Burnard. 2007. Reference Guide for the British National Corpus. http://www.natcorp.ox.ac.uk/docs/URG/.

Stephen Clark and James R. Curran. 2007. Wide-Coverage Efficient Statistical Parsing with CCG and Log-Linear Models. *Computational Linguistics*, 33(4):493—552.

Stephen Clark. 2014. Vector Space Models of Lexical Meaning (to appear). In Shalom Lappin and Chris Fox, editors, *Handbook of Contemporary Semantics*. Wiley-Blackwell, Oxford.

James R. Curran and Marc Moens. 2002a. Improvements in Automatic Thesaurus Extraction. In *Proceedings of the ACL-02 workshop on Unsupervised lexical acquisition-Volume 9*, pages 59–66. Association for Computational Linguistics.

James R. Curran and Marc Moens. 2002b. Scaling Context Space. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 231–238. Association for Computational Linguistics.

James R. Curran. 2004. *From Distributional to Semantic Similarity*. Ph.D. thesis, University of Edinburgh.

Katrin Erk. 2012. Vector Space Models of Word Meaning and Phrase Meaning: A Survey. *Language and Linguistics Compass*, 6(10):635–653.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing Search in Context: The Concept Revisited. *ACM Transactions on Information Systems*, 20(1):116—131.

Yoav Goldberg and Jon Orwant. 2013. A Dataset of Syntactic-Ngrams over Time from a Very Large Corpus of English Books. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 241–247, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Gregory Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, Norwell, MA, USA.

Z. Harris. 1954. Distributional Structure. *Word*, 10(23):146—162.

F. Hill, D. Kiela, and A. Korhonen. 2013. Concreteness and Corpora: A Theoretical and Practical Analysis. In *Proceedings of ACL 2013, Workshop on Cognitive Modelling and Computational Linguistics*, Sofia, Bulgaria.

Thomas K. Landauer and Susan T. Dumais. 1997. A solution to Platos problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240.

Gabriella Lapesa and Stefan Evert. 2013. Evaluating neighbor rank and distance measures as predictors of semantic priming. In *In Proceedings of the ACL Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2013)*, Sofia, Bulgaria.

G.A. Miller and W.G. Charles. 1991. Contextual Correlates of Semantic Similarity. *Language and Cognitive Processes*, 6(1):1—28.

Jeff Mitchell and Mirella Lapata. 2010. Composition in Distributional Models of Semantics. *Cognitive Science*, 34(8):1388–1429.

Sebastian Pado and Mirella Lapata. 2007. Dependency-based Construction of Semantic Space Models. *Computational Linguistics*, 33(2):161–199.

Douglas L. T. Rohde, Laura M. Gonnerman, and David C. Plaut. 2006. An Improved Model of Semantic Similarity based on Lexical Co-occurence. *Communciations of the ACM*, 8:627–633.

Herbert Rubenstein and John B. Goodenough. 1965. Contextual Correlates of Synonymy. *Commun. ACM*, 8(10):627–633, October.

Magnus Sahlgren. 2006. *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. Ph.D. thesis, Department of Linguistics, Stockholm University.

Sabine Schulte im Walde, Stefan Müller, and Stephen Roller. 2013. Exploring Vector Space Models to Predict the Compositionality of German Noun-Noun Compounds. In *Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics*, pages 255–265, Atlanta, GA.

N. Clayton Silver and William P. Dunlap. 1987. Averaging Correlation Coefficients: Should Fisher's z Transformation Be Used? *Journal of Applied Psychology*, 72(1):146–148, February.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press, Cambridge, MA, USA.

Benjamin P. Stone, Simon J. Dennis, and Peter J. Kwantes. 2008. A Systematic Comparison of Semantic Models on Human Similarity Rating Data: The Effectiveness of Subspacing. In *The Proceedings of the Thirtieth Conference of the Cognitive Science Society*.

Peter D. Turney and Patrick Pantel. 2010. From Frequency to Meaning: vector space models of semantics. *J. Artif. Int. Res.*, 37(1):141–188, January.

Julie Weeds, David Weir, and Diana McCarthy. 2004. Characterising Measures of Lexical Distributional Similarity. In *Proceedings of Coling 2004*, pages 1015–1021, Geneva, Switzerland, Aug 23–Aug 27. COLING.

S. Zelikovitz and M. Kogan. 2006. Using Web Searches on Important Words to create Background Sets for LSI Classification. In *In Proceedings of the 19th International FLAIRS Conference*, pages 598—603, Menlo Park, CA. AAAI Press.