View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Waseda University Repository

Testing APSyn against Vector Cosine on Similarity Estimation

Enrico Santus[1], Emmanuele Chersoni[2], Alessandro Lenci[3], Chu-Ren Huang[1], Philippe Blache[2]
[1] The Hong Kong Polytechnic University, Hong Kong
[2] Aix-Marseille University
[3] University of Pisa
{esantus, emmanuelechersoni}@gmail.com
alessandro.lenci@unipi.it
churen.huang@polyu.edu.hk
blache@lpl-aix.fr

Abstract

In Distributional Semantic Models (DSMs), Vector Cosine is widely used to estimate similarity between word vectors, although this measure was noticed to suffer from several shortcomings. The recent literature has proposed other methods which attempt to mitigate such biases. In this paper, we intend to investigate APSyn, a measure that computes the extent of the intersection between the most associated contexts of two target words, weighting it by context relevance. We evaluated this metric in a similarity estimation task on several popular test sets, and our results show that APSyn is in fact highly competitive, even with respect to the results reported in the literature for word embeddings. On top of it, APSyn addresses some of the weaknesses of Vector Cosine, performing well also on genuine similarity estimation.

1 Introduction

Word similarity is one of the most important and most studied problems in Natural Language Processing (NLP), as it is fundamental for a wide range of tasks, such as *Word Sense Disambiguation* (WSD), *Information Extraction* (IE), *Paraphrase Generation* (PG), as well as the automatic creation of semantic resources. Most of the current approaches to word similarity estimation rely on some version of the Distributional Hypothesis (DH), which claims that words occurring in the same contexts tend to have similar meanings (Harris, 1954; Firth, 1957; Sahlgren, 2008). Such hypothesis provides the theoretical ground for Distributional Semantic Models (DSMs), which represent word meaning by means of high-dimensional vectors encoding corpus-extracted co-occurrences between targets and their linguistic contexts (Turney and Pantel, 2010).

Traditional DSMs initialize vectors with co-occurrence frequencies. Statistical measures, such as Positive Pointwise Mutual Information (PPMI) or its variants (Church and Hanks, 1990; Bullinaria and Levy, 2012; Levy et al., 2015), have been adopted to normalize these values. Also, these models have exploited the power of dimensionality reduction techniques, such as Singular Value Decomposition (SVD; Landauer and Dumais, 1997) and Random Indexing (Sahlgren, 2005).

These first-generation models are currently referred to as count-based, as distinguished from the context-predicting ones, which have been recently proposed in the literature (Bengio et al., 2006; Collobert and Weston, 2008; Turian et al., 2010; Huang et al., 2012; Mikolov et al., 2013). More commonly known as *word embeddings*, these second-generation models learn meaning representations through neural network training: the vectors dimensions are set to maximize the probability for the contexts that typically occur with the target word.

Vector Cosine is generally adopted by both types of models as a similarity measure. However, this metric has been found to suffer from several problems (Li and Han, 2013; Faruqui et al., 2016), such as a bias towards features with higher values and the inability of considering how many features are actually shared by the vectors. Finally, Cosine is affected by the hubness effect (Dinu et al., 2014; Schnabel et al., 2015), i.e. the fact that words with high frequency tend to be universal neighbours. Even though other measures have been proposed in the literature (Deza and Deza, 2009), Vector Cosine is still by far the most popular one (Turney and Pantel, 2010). However, in a recent paper of Santus et al. (2016b), the authors have claimed that Vector Cosine is outperformed by APSyn (Average Precision for Synonymy), a metric based on the extent of the intersection between the most salient contexts of two target words. The measure, tested on a window-based DSM, outperformed Vector Cosine on the ESL and on the TOEFL datasets.

In the present work, we perform a systematic evaluation of APSyn, testing it on the most popular test sets for similarity estimation - namely WordSim-353 (Finkelstein et al., 2001), MEN (Bruni et al., 2014) and SimLex-999 (Hill et al., 2015). For comparison, Vector Cosine is also calculated on several count-based DSMs. We implement a total of twenty-eight models with different parameters settings, each of which differs according to corpus size, context window width, weighting scheme and SVD application. The new metric is shown to outperform Vector Cosine in most settings, except when the latter metric is applied on a PPMI-SVD reduced matrix (Bullinaria and Levy, 2012), against which APSyn still obtains competitive performances. The results are also discussed in relation to the state-of-the-art DSMs, as reported in Hill et al. (2015). In such comparison, the best settings of our models outperform the word embeddings in almost all datasets. A pilot study was also carried out to investigate whether APSyn is scalable. Results prove its high performance also when calculated on large corpora, such as those used by Baroni et al. (2014).

On top of the performance, APSyn seems not to be subject to some of the biases that affect Vector Cosine. Finally, considering the debate about the ability of DSMs to calculate genuine similarity as opposed to word relatedness (Turney, 2001; Agirre et al., 2009; Hill et al., 2015), we test the ability of the models to quantify genuine semantic similarity.

2 Background

2.1 DSMs, Measures of Association and Dimensionality Reduction

Count-based DSMs are built in an unsupervised way. Starting from large preprocessed corpora, a matrix $M_{(m \times n)}$ is built, in which each row is a vector representing a target word in a vocabulary of size *m*, and each column is one of the *n* potential contexts (Turney and Pantel, 2010; Levy et al., 2015). The vector dimensions are counters recording how many times the contexts co-occur with the target words.

Since raw frequency is highly skewed, most DSMs have adopted more sophisticated association measures, such as Positive PMI (PPMI; Church and Hanks, 1990; Bullinaria and Levy, 2012; Levy et al., 2015) and Local Mutual Information (LMI; Evert, 2005). PPMI compares the observed joint probability of co-occurrence of *w* and *c* with their probability of co-occurrence assuming statistical indipendence. It is defined as:

$$PPMI(w,c) = max(PMI(w,c), 0) \tag{1}$$

$$PMI(w,c) = log\left(\frac{P(w,c)}{P(w)P(c)}\right) = log\left(\frac{|w,c|D}{|w||c|}\right) \tag{2}$$

where *w* is the target word, *c* is the given context, *P(w,c)* is the probability of co-occurrence, and *D* is the collection of observed word-context pairs.

Unlike frequency, PPMI was found to have a bias towards rare events. LMI could therefore be used to reduce such bias and it consists in multiplying the PPMI of the pair by its co-occurrence frequency. Since target words may occur in hundreds of thousands contexts, most of which are not informative, methods for dimensionality reduction have been investigated, such as truncated SVD (Deerwester et al., 1990; Landauer and Dumais, 1997; Turney and Pantel, 2010; Levy et al., 2015). SVD has been regarded as a method for noise reduction and for the discovery of latent dimensions of meaning, and it has been shown to improve similarity measurements when combined with PPMI (Bullinaria and Levy, 2012; Levy et al., 2015). As we will see in the next section, APSyn applies another type of reduction, which consists in selecting only the top-ranked contexts in a relevance sorted context list for each word vector. Such reduction complies with the principle of cognitive economy (i.e. only the most relevant contexts are elaborated; see Finton, 2002) and with the results of behavioural studies, which supported feature saliency (Smith et al., 1974). Since APSyn was defined for linguistic contexts (Santus et al., 2016b), we did not test it on SVD-reduced spaces, leaving such test to further studies.

2.2 Similarity Measures

Vector Cosine, by far the most common distributional similarity metric (Turney and Pantel, 2010; Landauer and Dumais, 1997; Jarmasz and Szpakowicz, 2004; Mikolov et al., 2013; Levy et al., 2015), looks at the normalized correlation between the dimensions of two word vectors, w_1 and w_2 and returns a score between -1 and 1. It is described by the following equation:

$$cos(w_1, w_2) = \frac{\sum_{i=1}^{n} f_1 i \times f_{2i}}{\sqrt{\sum_{i=1}^{n} f_1 i} \times \sqrt{\sum_{i=1}^{n} f_2 i}} \quad (3)$$

where $f_i x$ is the *i*-th dimension in the vector *x*.

Despite its extensive usage, Vector Cosine has been recently criticized for its hyper sensibility to features with high values and for the inability of identifying the actual feature intersection (Li and Han, 2013; Schnabel et al., 2015). Recalling an example by Li and Han (2013), the Vector Cosine for the toy-vectors $a = [1, 2, 0]$ and $b = [0, 1, 0]$ (i.e. 0.8944) is unexpectedly higher than the one for a and $c = [2, 1, 0]$ (i.e. 0.8000), and even higher than the one for the toy-vectors *a* and $d = [1, 2, 1]$ (i.e. 0.6325), which instead share a larger feature intersection. Since the Vector Cosine is a distance measure, it is also subject to the hubness problem, which was shown by Radovanovic et al. (2010) to be an inherent property of data distributions in high-dimensional vector space. The problem consists in the fact that vectors with high frequency tend to get high scores with a large number of other vectors, thus becoming universal nearest neighbours (Dinu et al., 2014; Schnabel et al., 2015; Faruqui et al., 2016).

Another measure of word similarity named APSyn [1] has been recently introduced in Santus et al. (2016a) and Santus et al. (2016b), and it was shown to outperform the vector cosine on the TOEFL (Landauer and Dumais, 1997) and on the ESL (Turney, 2001) test sets. This measure is based on the hypothesis that words carrying similar meanings share their most relevant contexts in higher proportion compared to less similar words. The authors define APSyn as the extent of the weighted intersection between the top most salient contexts of the target words, weighting it by the average rank of the intersected features in the PPMI-sorted contexts lists of the target words:

$$APSyn(w_1, w_2) = \sum_{f \epsilon N(F_1) \cap N(F_2)} \frac{1}{(rank_1(f) + rank_2(f))/2} \quad (4)$$

meaning: for every feature *f* included in the intersection between the top *N* features of w_1 and the top of w_2 (i.e. $N(f_1)$ and $N(f_2)$), add 1 divided by the average rank of the feature in the PPMI-ranked features of w_1 (i.e. $rank_1$) and w_2 (i.e. $rank_2$). According to the authors, *N* is a parameter, generally ranging between 100 and 1000. Results are shown to be relatively stable when *N* varies in this range, while become worst if bigger *N* are used, as low informative features are also introduced. Santus et al. (2016a) have also used LMI instead of PPMI as weighting function, but achieving lower results.

With respect to the limitations mentioned above for the Vector Cosine, APSyn has some advantages. First of all, it is by definition able to identify the extent of the intersection. Second, its sensibility to features with high values can be kept under control by tuning the value of *N*. On top of it, feature values (i.e. their weights) do not affect directly the similarity score, as they are only used to build the feature rank. With reference to the toy-vectors presented above, APSyn would assign in fact completely different scores. The higher score would be assigned to a and *d*, as they share two relevant features out of three. The second higher score would be assigned to a and *c*, for the same reason as above. The lower score would be instead assigned to a and b, as they only share one non-salient feature. In section 3.4, we briefly discuss the hubness problem.

[1] Scripts and information can be found at https://github.com/esantus/APSyn

2.3 Datasets

For our evaluation, we used three widely popular datasets: WordSim-353 (Finkelstein et al., 2001), MEN (Bruni et al., 2014), SimLex-999 (Hill et al., 2015). These datasets have a different history, but all of them consist in word pairs with an associated score, that should either represent word association or word similarity. WordSim-353 (Finkelstein et al., 2001) was proposed as a word similarity dataset containing 353 pairs annotated with scores between 0 and 10. However, Hill et al. (2015) claimed that the instructions to the annotators were ambiguous with respect to similarity and association, so that the subjects assigned high similarity scores to entities that are only related by virtue of frequent association (e.g. *coffee* and *cup*; *movie* and *theater*). On top of it, WordSim-353 does not provide the POS-tags for the 439 words that it contains, forcing the users to decide which POS to assign to the ambiguous words (e.g. [*white*, *rabbit*] and [*run*, *marathon*]). An extension of this dataset resulted from the subclassification carried out by Agirre et al. (2009), which discriminated between similar and associated word pairs. Such discrimination was done by asking annotators to classify all pairs according to the semantic relation they hold (i.e. identical, synonymy, antonymy, hypernymy, meronymy and none-of-the-above). The annotation was then used to group the pairs in three categories: similar pairs (those classified as identical, synonyms, antonyms and hypernyms), associated pairs (those classified as meronyms and none-of-the-above, with an average similarity greater than 5), and non-associated pairs (those classified as none-of-the-above, with an average similarity below or equal to 5). Two gold standard were finally produced: i) one for similarity, containing 203 word pairs resulting from the union of similar and non-associated pairs; ii) one for relatedness, containing 252 word pairs resulting from the union of associated and non-associated pairs. Even though such a classification made a clear distinction between the two types of relations (i.e. similarity and association), Hill et al. (2015) argue that these gold standards still carry the scores they had in WordSim-353, which are known to be ambiguous in this regard.

The MEN Test Collection (Bruni et al., 2014) includes 3,000 word pairs divided in two sets (one for training and one for testing) together with human judgments, obtained through Amazon Mechanical Turk. The construction was performed by asking subjects to rate which pair - among two of them - was the more related one (i.e. the most associated). Every pairs-couple was proposed only once, and a final score out of 50 was attributed to each pair, according to how many times it was rated as the most related. According to Hill et al. (2015), the major weakness of this dataset is that it does not encode word similarity, but a more general notion of association.

SimLex-999 is the dataset introduced by Hill et al. (2015) to address the above mentioned criticisms of confusion between similarity and association. The dataset consists of 999 pairs containing 1,028 words, which were also evaluated in terms of POS-tags and concreteness. The pairs were annotated with a score between 0 and 10, and the instructions were strictly requiring the identification of word similarity, rather than word association. Hill et al. (2015) claim that differently from other datasets, SimLex-999 inter-annotator agreement has not been surpassed by any automatic approach.

2.4 State of the Art Vector Space Models

In order to compare our results with state-of-the-art DSMs, we report the scores for the Vector Cosines calculated on the neural language models (NLM) by Hill et al. (2015), who used the code (or directly the embeddings) shared by the original authors. As we trained our models on almost the same corpora used by Hill and colleagues, the results are perfectly comparable.

The three models we compare our results to are: i) the convolutional neural network of Collobert and Weston (2008), which was trained on 852 million words of Wikipedia; ii) the neural network of Huang et al. (2012), which was trained on 990 million words of Wikipedia; and iii) the word2vec of Mikolov et al. (2013), which was trained on 1000 million words of Wikipedia and on the RCV Vol. 1 Corpus (Lewis et al., 2004).

Dataset	**SimLex-999**		**WordSim-353**		**MEN**	
Window	**2**	**3**	**2**	**3**	**2**	**3**
Cos Freq	0.149	0.133	0.172	0.148	0.089	0.096
Cos LMI	0.248	0.259	0.321	0.32	0.336	0.364
Cos PPMI	0.284	0.267	0.41	0.407	0.424	0.433
Cos SVD-Freq300	0.128	0.127	0.169	0.172	0.076	0.084
Cos SVD-LMI300	0.19	0.21	0.299	0.29	0.275	0.286
Cos SVD-PPMI300	**0.386**	**0.382**	**0.485**	**0.47**	**0.509**	**0.538**
APSynLMI-1000	0.18	0.163	0.254	0.237	0.205	0.196
APSynLMI-500	0.199	0.164	0.283	0.265	0.226	0.214
APSynLMI-100	0.206	0.182	0.304	0.265	0.23	0.209
APSynPPMI-1000	0.254	0.304	0.399	0.453	0.369	0.415
APSynPPMI-500	0.295	0.32	**0.455**	**0.468**	0.423	0.478
APSynPPMI-100	**0.332**	**0.328**	0.425	0.422	**0.481**	**0.513**
State of the Art						
Mikolov et al.	0.282		0.442		0.433	

Table 1: Spearman correlation scores for our eight models trained on RCV Vol. 1, in the three datasets Simlex-999, WordSim-353 and MEN. In the bottom the performance of the state-of-the-art model of Mikolov et al. (2013), as reported in Hill et al. (2015).

3 Experiments

In this section, we describe our experiments, starting from the training corpora (Section 3.1), to move to the implementation of twenty-eight DSMs (Section 3.2), following with the application and evaluation of the measures (Section 3.3), up to the performance analysis (Section 3.4) and the scalability test (Section 3.5).

3.1 Corpora and Preprocessing

We used two different corpora for our experiments: RCV vol. 1 (Lewis et al., 2004) and the Wikipedia corpus (Baroni et al., 2009), respectively containing 150 and 820 million words. The RCV Vol. 1 and Wikipedia were automatically tagged, respectively, with the POS tagger described in Dell'Orletta (2009) and with the TreeTagger (Schmid, 1994).

3.2 DSMs

For our experiments, we implemented twenty-eight DSMs, but for reasons of space only sixteen of them are reported in the tables. All of them include the pos-tagged target words used in the three datasets (i.e. MEN, WordSim-353 and SimLex-999) and the pos-tagged contexts having frequency above 100 in the two corpora. We considered as contexts the content words (i.e. nouns, verbs and adjectives) within a window of 2, 3 and 5, even though the latter was given up for its poor performances.

As for SVD factorization, we found out that the best results were always achieved when the number of latent dimensions was between 300 and 500. We report here only the scores for $k = 300$, since 300 is one of the most common choices for the dimensionality of SVD-reduced spaces and it is always close to be an optimal value for the parameter.

Fourteen out of twenty-eight models were developed for RCV1, while the others were developed for Wikipedia. For each corpus, the models differed according to the window size (i.e. 2 and 3), to the statistical association measure used as a weighting scheme (i.e. none, PPMI and LMI) and to the application of SVD to the previous combinations.

3.3 Measuring Word Similarity and Relatedness

Given the twenty-eight DSMs, for each dataset we have measured the Vector Cosine and APSyn between the words in the test pairs.

Dataset	**SimLex-999**		**WordSim-353**		**MEN**	
Window	**2**	**3**	**2**	**3**	**2**	**3**
Cos Freq	0.148	0.159	0.199	0.207	0.178	0.197
Cos LMI	0.367	0.374	0.489	0.529	0.59	0.63
Cos PPMI	0.395	0.364	0.605	0.622	0.733	0.74
Cos SVD-Freq300	0.157	0.184	0.159	0.172	0.197	0.226
Cos SVD-LMI300	0.327	0.329	0.368	0.408	0.524	0.563
Cos SVD-PPMI300	**0.477**	**0.464**	**0.533**	**0.562**	**0.769**	**0.779**
APSynLMI-1000	0.343	0.344	0.449	0.477	0.586	0.597
APSynLMI-500	0.339	0.342	0.438	0.47	0.58	0.588
APSynLMI-100	0.303	0.31	0.392	0.428	0.48	0.498
APSynPPMI-1000	0.434	0.419	0.599	0.643	0.749	0.772
APSynPPMI-500	**0.442**	**0.423**	**0.602**	**0.653**	**0.757**	**0.773**
APSynPPMI-100	0.316	0.281	0.58	0.608	0.703	0.722
State of the Art						
Huang et al.	0.098		0.3		0.433	
Collobert & Weston	0.268		0.494		0.575	
Mikolov et al.	0.414		0.655		0.699	

Table 2: Spearman correlation scores for our eight models trained on Wikipedia, in the three datasets Simlex-999, WordSim-353 and MEN. In the bottom the performance of the state-of-the-art models of Collobert and Weston (2008), Huang et al. (2012), Mikolov et al. (2013), as reported in Hill et al. (2015).

The Spearman correlation between our scores and the gold standard was then computed for every model and it is reported in Table 1 and Table 2. In particular, Table 1 describes the performances on SimLex-999, WordSim-353 and MEN for the measures applied on RCV Vol. 1 models. Table 2, instead, describes the performances of the measures on the three datasets for the Wikipedia models. Concurrently, Table 3 and Table 4 describe the performances of the measures respectively on the RCV Vol. 1 and Wikipedia models, tested on the subsets of WordSim-353 extracted by Agirre et al. (2009).

3.4 Performance Analysis

Table 1 shows the Spearman correlation scores for Vector Cosine and APSyn on the three datasets for the eight most representative DSMs built using RCV Vol. 1. Table 2 does the same for the DSMs built using Wikipedia. For the sake of comparison, we also report the results of the state-of-the-art DSMs mentioned in Hill et al. (2015) (see Section 2.5).

With a glance at the tables, it can be easily noticed that the measures perform particularly well in two models: i) APSyn, when applied on the PPMI-weighted DSM (henceforth, APSynPPMI); ii) Vector Cosine, when applied on the SVD-reduced PPMI-weighted matrix (henceforth, CosSVDPPMI). These two models perform consistently and in a comparable way across the datasets, generally outperforming the state-of-the-art DSMs, with an exception for the Wikipedia-trained models in WordSim-353.

Some further observations are: i) corpus size strongly affects the results; ii) PPMI strongly outperforms LMI for both Vector Cosine and APSyn; iii) SVD boosts the Vector Cosine, especially when it is combined with PPMI; iv) *N* has some impact on the performance of APSyn, which generally achieves the best results for *N*=500. As a note about iii), the results of using SVD jointly with LMI spaces are less predictable than when combining it with PPMI.

Also, we can notice that the smaller window (i.e. 2) does not always perform better than the larger one (i.e. 3). The former appears to perform better on SimLex-999, while the latter seems to have some advantages on the other datasets. This

Dataset	**WSim (SIM)**		**WSim (REL)**	
Window	**2**	**3**	**2**	**3**
Cos Freq	0.208	0.158	0.167	0.175
Cos LMI	0.416	0.395	0.251	0.269
Cos PPMI	0.52	0.496	0.378	0.396
Cos SVD-Freq300	0.240	0.214	0.051	0.084
Cos SVD-LMI300	0.418	0.393	0.141	0.151
Cos SVD-PPMI300	**0.550**	**0.522**	**0.325**	**0.323**
APSynLMI-1000	0.32	0.29	0.259	0.241
APSynLMI-500	0.355	0.319	0.261	0.284
APSynLMI-100	0.388	0.335	0.233	0.27
APSynPPMI-1000	0.519	0.525	0.337	**0.397**
APSynPPMI-500	**0.564**	**0.546**	**0.361**	0.382
PMI APSynPPMI-100	0.562	0.553	0.287	0.309

Table 3: Spearman correlation scores for our eight models trained on RCV1, in the two subsets of WordSim-353.

might depend on the different type of similarity encoded in SimLex-999 (i.e. genuine similarity). On top of it, despite Hill et al. (2015)'s claim that no evidence supports the hypothesis that smaller context windows improve the ability of models to capture similarity (Agirre et al., 2009; Kiela and Clark, 2014), we need to mention that window 5 was abandoned because of its low performance.

With reference to the hubness effect, we have conducted a pilot study inspired to the one carried out by Schnabel et al. (2015), using the words of the SimLex-999 dataset as query words and collecting for each of them the top 1000 nearest neighbors. Given all the neighbors at rank *r*, we have checked their rank in the frequency list extracted from our corpora. Figure 1 shows the relation between the rank in the nearest neighbor list and the rank in the frequency list. It can be easily noticed that the highest ranked nearest neighbors tend to have higher rank also in the frequency list, supporting the idea that frequent words are more likely to be nearest neighbors. APSyn does not seem to be able to overcome such bias, which seems to be in fact an inherent property of the DSMs (Radovanovic et al., 2010). Further investigation is needed to see whether variations of APSyn can tackle this problem.

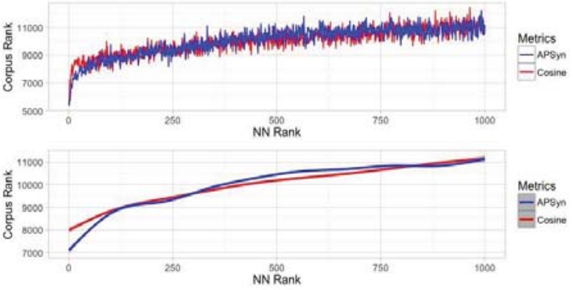

Figure 1: Rank in the corpus-derived frequency list for the top 1000 nearest neighbors of the terms in SimLex-999, computed with Cosine (red) and APSyn (blue). The smoothing chart in the bottom uses the Generalized Additive Model (GAM) from the *mgcv* package in *R*.

Finally, few words need to be spent with regard to the ability of calculating genuine similarity, as distinguished from word relatedness (Turney, 2001; Agirre et al., 2009; Hill et al., 2015). Table 3 and Table 4 show the Spearman correlation scores for the two measures calculated on the models respectively trained on RCV1 and Wikipedia, tested on the subsets of WordSim-353 extracted by Agirre et al. (2009). It can be easily noticed that our best models work better on the similarity subset. In particular, APSynPPMI performs about 20-30% better for the similarity subset than for the relatedness one (see Table 3), as well as both APSynPPMI and CosSVDPPMI do in Wikipedia (see Table 4).

Dataset	**WSim (SIM)**		**WSim (REL)**	
Window	**2**	**3**	**2**	**3**
Cos Freq	0.335	0.334	0.03	0.05
Cos LMI	0.638	0.663	0.293	0.34
Cos PPMI	0.672	0.675	0.441	0.446
Cos SVD-Freq300	0.35	0.363	-0.013	0.001
Cos SVD-LMI300	0.604	0.626	0.222	0.286
Cos SVD-PPMI300	**0.72**	**0.725**	**0.444**	**0.486**
APSynLMI-1000	0.609	0.609	0.317	0.36
APSynLMI-500	0.599	0.601	0.289	0.344
APSynLMI-100	0.566	0.574	0.215	0.271
APSynPPMI-1000	0.692	0.726	0.507	0.568
APSynPPMI-500	**0.699**	**0.742**	**0.508**	**0.571**
APSynPPMI-100	0.66	0.692	0.482	0.516

Table 4: Spearman correlation results for our eight models trained on Wikipedia, in the subsets of WordSim-353.

3.5 Scalability

In order to evaluate the scalability of APSyn, we have performed a pilot test on WordSim-353 and MEN with the same corpus used by Baroni et al. (2014), which consists of about 2.8B words (i.e. about 3 times Wikipedia and almost 20 times RCV1). The best scores were obtained with APSyn, *N*=1000, on a 2-window PPMI-weighted DSM. In such setting, we obtain a Spearman correlation of 0.72 on WordSim and 0.77 on MEN. These results are much higher than those reported by Baroni et al. (2014) for the count-based models (i.e. 0.62 on WordSim and 0.72 on MEN) and slightly lower than those reported for the predicting ones (i.e. 0.75 on WordSim and 0.80 on MEN).

4 Conclusions

In this paper, we have presented the first systematic evaluation of APSyn, comparing it to Vector Cosine in the task of word similarity identification. We developed twenty-eight count-based DSMs, each of which implementing different hyperparameters. PPMI emerged as the most efficient association measure: it works particularly well with Vector Cosine, when combined with SVD, and it boosts APSyn.

APSyn showed extremely promising results, despite its conceptual simplicity. It outperforms the Vector Cosine in almost all settings, except when the latter is used on a PPMI-weighed SVD-reduced DSM. Even in this case, anyway, its performance is very competitive. Interestingly, our best models achieve results that are comparable to - or even better than - those reported by Hill et al. (2015) for the state-of-the-art word embeddings models. In Section 3.5 we show that APSyn is scalable, outperforming the state-of-the-art count-based models reported in Baroni et al. (2014). On top of it, APSyn does not suffer from some of the problems reported for the Vector Cosine, such as the inability of identifying the number of shared features. It still however seems to be affected by the hubness issue, and more research should be carried out to tackle it. Concerning the discrimination between similarity and association, the good performance of APSyn on SimLex-999 (which was built with a specific attention to genuine similarity) and the large difference in performance between the two subsets of WordSim-353 described in Table 3 and Table 4 make us conclude that APSyn is indeed efficient in quantifying genuine similarity.

To conclude, being a linguistically and cognitively grounded metric, APSyn offers the possibility for further improvements, by simply combining it to other properties that were not yet considered in its definition. A natural extension would be to verify whether APSyn hypothesis and implementation holds on SVD reduced matrices and word embeddings.

Acknowledgments

This paper is partially supported by HK PhD Fellowship Scheme, under PF12-13656. Emmanuele Chersoni's research is funded by a grant of the University Foundation A*MIDEX. Thanks to Davis Ozols for the support with *R*.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27. Association for Computational Linguistics.

Luis Von Ahn and Laura Dabbish. 2004. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326. ACM.

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

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, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *ACL (1)*, pages 238–247.

Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. 2006. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *J. Artif. Intell. Res.(JAIR)*, 49(1-47).

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

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

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391.

Felice DellOrletta. 2009. Ensemble system for part-of-speech tagging. *Proceedings of EVALITA*, 9.

Michel Marie Deza and Elena Deza. 2009. *Encyclopedia of distances*. Springer.

Georgiana Dinu, Angeliki Lazaridou and Marco Baroni 2014. Improving zero-shot learning by mitigating the hubness problem. *arXiv preprint arXiv:1603.09054*.

Stefan Evert. 2005. The statistics of word cooccurrences: word pairs and collocations.

Manaal Faruqui, Yulia Tsvetkov, Pushpendre Ratogi, and Chris Dyer. 2016. Problems With Evaluation of Word Embeddings Using Word Similarity Tasks. *arXiv preprint arXiv:1301.3781.*.

Christiane Fellbaum. 1998. *WordNet*. Wiley Online Library.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414. ACM.

David Finton. 2002. Cognitive economy and the role of representation in on-line learning. Doctoral dissertation. University of Wisconsin-Madison.

John Rupert Firth. 1957. *Papers in linguistics, 1934-1951*. Oxford University Press.

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

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*.

Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics.

Mario Jarmasz and Stan Szpakowicz. 2004. Rogets thesaurus and semantic similarity1. *Recent Advances in Natural Language Processing III: Selected Papers from RANLP*, 2003:111.

Douwe Kiela and Stephen Clark. 2014. A systematic study of semantic vector space model parameters. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC) at EACL*, pages 21–30.

Thomas K Landauer and Susan T Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211.

Thomas K Landauer, Peter W Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research*, 5:361–397.

Baoli Li and Liping Han. 2013. Distance weighted cosine similarity measure for text classification.. *Intelligent Data Engineering and Automated Learning -IDEAL 2013*: 611-618.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Milos Radovanovic, Alexandros Nanopoulos and Mirjana Ivanovic. 2010. On the existence of obstinate results in vector space models. *Proceedings of SIGIR*:186-193.

Magnus Sahlgren. 2005. An introduction to random indexing. In *Methods and applications of semantic indexing workshop at the 7th international conference on terminology and knowledge engineering, TKE*, volume 5.

Magnus Sahlgren. 2008. The distributional hypothesis. *Italian Journal of Linguistics*, 20(1):33–54.

Enrico Santus, Tin-Shing Chiu, Qin Lu, Alessandro Lenci, and Chu-ren Huang. 2016. Unsupervised Measure of Word Similarity: How to Outperform Co-Occurrence and Vector Cosine in VSMs. *arXiv preprint arXiv:1603.09054*.

Enrico Santus, Tin-Shing Chiu, Qin Lu, Alessandro Lenci, and Chu-ren Huang. 2016. What a Nerd! Beating Students and Vector Cosine in the ESL and TOEFL Datasets. In *Proceedings of LREC*.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the international conference on new methods in language processing*, volume 12, pages 44–49. Citeseer.

Tobias Schnabel, Igor Labutov, David Mimmo and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of EMNLP*.

Edward Smith, Edward Shoben and Lance Rips. 1974. Structure and process in semantic memory: A featural model for semantic decisions. *Psychological Review*, 81(3).

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

Peter D Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188.

Peter D Turney. 2001. Mining the web for synonyms: Pmi-ir versus lsa on toefl.