Noname manuscript No.
(will be inserted by the editor)

Comparing Explicit and Predictive Distributional Semantic Models Endowed with Syntactic Contexts

Pablo Gamallo

Received: date / Accepted: date

Abstract In this article, we introduce an explicit count-based strategy to build word space models with syntactic contexts (dependencies). A filtering method is defined to reduce explicit word-context vectors. This traditional strategy is compared with a neural embedding (predictive) model also based on syntactic dependencies. The comparison was performed using the same parsed corpus for both models. Besides, the dependency-based methods are also compared with bag-of-words strategies, both count-based and predictive ones. The results show that our traditional count-based model with syntactic dependencies outperforms other strategies, including dependency-based embeddings, but just for the tasks focused on discovering similarity between words with the same function (i.e. near-synonyms).

1 Introduction

The existing distributional methods for estimating word similarity rely on the old observation that semantically related words tend to occur in similar contexts (Harris, 1985). These methods differ in, at least, three important aspects: the type of context (e.g., bag-of-words, syntactic dependencies), the similarity measure, and the way the word space model is built: count-based, singular value decomposition (SVD), embeddings, etc.

In traditional word space models, word distributions are defined as high dimensional but sparse vectors. As raw co-occurrence counts do not always work well, distributional semantic models seem to achieve higher performance when various transformations are applied to the raw vectors, for example by reweighing the counts for context informativeness and smoothing them

Pablo Gamallo
Centro Singular de Investigación en Tecnoloxías da Información (CiTIUS)
University of Santiago de Compostela, Galiza, Spain
E-mail: pablo.gamallo@usc.es
htpp://gramatica.usc.es/~gamallo

with dimensionality reduction techniques. Dimensionality can be reduced using techniques such as singular value decomposition (SVD) (Landauer and Dumais, 1997) or principal component analysis (PCA) (Lebret and Collobert, 2015). These techniques give rise to dense vectors. More recently, other kind of dense vectors derived from neural-network language modeling have been proposed to represent words (Collobert and Weston, 2008; Mikolov et al, 2013). These dense representations are known as *word embeddings* or *predictive models*, while those using word-context co-occurrences are known as *explicit* or *count-based* models . There is some controversy about the performance of the different types of word space models when they are applied on specific NLP tasks. Some researchers claim that word spaces based on neural embeddings outperform traditional count-based models to compute word similarity (Baroni et al, 2014b; Mikolov et al, 2013). Other researchers, by contrast, show that there are no significant differences between them (Lebret and Collobert, 2015; Levy and Goldberg, 2014b; Levy et al, 2015), and claim that both embeddings and explicit models have actually succeeded in capturing word similarities. Other works report heterogeneous results since the performance of the two models varies according to the task to be evaluated (Blacoe and Lapata, 2012; Huang et al, 2012)

In addition to the quality of context information represented in the word space model, efficiency is also an important issue. Dense representations are considered to be easy to work with because they enable efficient computation of word similarities through low-dimensional matrix operations. Cosine similarity is one of the most used measures to compute similarity between dense vectors, including word embeddings. However, traditional sparse vectors can also be represented in an efficient way on the basis of hashing functions whose keys are words-contexts pairs and their values are non-zero scores (Gamallo and Bordag, 2011). Biemann et al (2013) argue that there is no need to explicitly model non-existing relations, which would be zeros in the vector representation; it is only worthwhile storing contexts of words if those same contexts would be explicitly represented (non-zero) in a sparse representation. Besides, in order to still reduce the word-context co-occurrences, there exist filtering strategies to only select the most relevant contexts for each word (Bordag, 2008; Padró et al, 2014). Similarity measures used on this type of traditional representations are, among others, Lin measure (Lin, 1998), Cosine, or Dice coefficient (Curran and Moens, 2002), which only require non-zero values to compute word similarity.

Concerning the type of context used to represent word distributions, there is a great number of previous works that evaluate and compare syntactic contexts (usually dependencies) with bag-of-words techniques (Grefenstette, 1993; Seretan and Wehrli, 2006; Padó and Lapata, 2007; Peirsman et al, 2007; Gamallo, 2008, 2009; Levy and Goldberg, 2014a). All of them state that syntax-based methods outperform bag-of-words techniques, in particular when the objective is to compute semantic similarity between functional equivalent words, such as detection of co-hyponym/hypernym word relations (i.e. near synonymy). Syntactic contexts yield functional similarities of a co-hyponym

nature. Other works, however, are less conclusive and seem to conclude that syntax-based models are at least not worse than bag-of-words strategies (Baroni and Lenci, 2010).

The objective of this article is to describe yet another count-based word space model, using syntactic dependencies and based on a filtering method to reduce non-zero values. Special attention will be paid to the definition of syntactic context and the filtering strategy. This model will be compared to the dependency-based embeddings described in Levy and Goldberg (2014a). Moreover, we will also compare our syntax-based strategy against bag-of-words models (both count-based and embeddings). The results of our experiments indicate that our count-based model clearly outperforms both neural dependency-based embeddings and bag-of-words models for the task of discovering synonyms (or near-synonyms) and immediate hypernyms. Further experiments focused on discovering analogies (also called *relational similarity*) will also be reported.

The count-based model we propose is not an original strategy. It is based on the distributional models described in our previous work (Gamallo and Bordag, 2011; Gamallo et al, 2005). More precisely, the idea of filtering contexts was taken from (Gamallo and Bordag, 2011), and the definition of syntactic contexts was introduced in (Gamallo et al, 2005). In these two papers, the proposed count-based model was compared to many other count-based strategies, including Latent Semantic Analysis (Landauer and Dumais, 1997) and Lin's strategy (Lin, 1998). Therefore, the main contribution of the current work is not to propose and describe a new distributional word vector space, but to compare its performance against other models (including syntax-based embeddings) in different datasets.

This paper is organized as follows. In the next section (2), we explore some work comparing word embeddings, traditional dense matrices, and explicit space models. Next, Section 3 introduces the more innovative aspects of our traditional approach. Then, four different experiments are reported in 4 and, finally, some conclusions are addressed in 5.

2 Background and Related Work

Levy and Goldberg (2014c) showed that a word embedding generated by the algorithm based on skip-gram and negative sampling is implicitly factorizing a word-context matrix, whose cells are the point-wise mutual information (PMI) of the respective word and context pairs. That is, the mathematical algorithms underlying embeddings are in fact very similar to those employed by count-based methods to do matrix factorization for dimensionality reduction (e.g. SVD). Such a discovery seems to prove that neural embeddings are doing something very similar to what the NLP community has been doing for about 20 years. This is in accordance with recent work suggesting that traditional count-based models can perform just as well as (or even outperform) neural

embedding methods on some NLP tasks (Lebret and Collobert, 2015; Levy and Goldberg, 2014b).

In addition, in previous work (Gamallo and Bordag, 2011), we showed that dense matrix representations derived from SVD are not more computational efficient than sparse matrices represented as hash tables with just non-zero co-occurrences. We also showed that dense low-dimensional vectors do not make better generalizations than explicit representations with just observed word-context pairs. Explicit count-based models with good context filters tend to outperform dense representations in word similarity tasks.

In cognitive psychology and neurosciences, some authors claim that word models do not require hidden or abstract factor representations that have no meaning by themselves (Hofmann and Jacobs, 2014). To keep the strength of fully-transparent symbolic representations, Hofmann et al (2011) used the direct word co-occurrences with loglikelihood word-context weight instead of dimension-reduced latent variables. It is not clear whether dimension reduction provides a significant advantage in predicting human performance (Bullinaria and Levy, 2007) or in extracting brain activation patterns (Bullinaria and Levy, 2013)

Moreover, neural word-embeddings are considered opaque, in the sense that it is hard to assign meanings to the dense dimensions. By contrast, explicit count-based models are easy to interpret: it is easy to explore the contexts that were selected to be most discriminating of particular words (Levy and Goldberg, 2014a). However, we must outline attractive features of word-embedding approaches, like implicitly performing dimensionality reduction, having an implementation that is easy to use, a learning step without manual annotation, and being able to efficiently scale up to process very large amounts of input data.

In the literature, little attention has been paid to context filtering within count-based approaches. Most traditional approaches mainly focused on converting sparse matrices into dense ones by dimensionality reduction (Landauer and Dumais, 1997). However, there exists some work reducing the raw matrix by means of simple filtering strategies aimed at selecting the most relevant contexts for each word (Bordag, 2008; Biemann et al, 2013; Padró et al, 2014). The word model we propose follows this filtering strategy.

3 A Count Based Method

3.1 Dependency Based Contexts

Word contexts can be derived from the dependency relations the words participate in (e.g. subject, direct object, modifier). To extract contexts from dependencies, we use the co-compositional methodology defined in (Gamallo et al, 2005) and also more recently in (Levy and Goldberg, 2014a). For a target word w related to a set of dependents $d_1, ..., d_k$ and to a head h (since each word is only dependent of only one head), we extract the contexts:

$$(d_1, \downarrow rel_1)..., (d_k, \downarrow rel_k), (h, \uparrow rel_h)$$

where $\downarrow rel$ is a type of dependency relation containing a specific dependent word, and $\uparrow rel$ stands for the inverse relation: a dependency containing a specific head. For instance, in *"Mary smiled"*, $(mary, \downarrow subject)$ is a context of the verb "smiled", while $(smile, \uparrow subject)$ is a context of "Mary". Besides, prepositions take part of the dependency labels. Before the context extraction, prepositions are joined to the head so as to subsume the preposition itself into the dependency label. For instance, in *"smiled at Bill"*, $(bill, \downarrow prep\_at)$ is a context of "smiled" while $(smile, \uparrow prep\_at)$ is the inverse context of "Bill". According to the work described in Baroni and Lenci (2010), this sort of contexts belong to the *word by link-word* vector space model where vectors are labeled with words and vector dimensions with tuples consisting in a relation and a lexical word.

3.2 Context Filtering

Given the power-law distribution of words in a corpus, all co-occurrence matrices representing lexical knowledge are sparse. When storing and manipulating large sparse matrices on a computer, it is beneficial and often necessary to use specialized data structures that take advantage of the sparseness. Many if not most entries of a sparse matrix are zeros that do not need to be stored explicitly. A possible storage mode for a sparse matrix is a hash table where keys are word-context pairs with non-zero values (Gamallo and Bordag, 2011).

To reduce the number of keys in a hash table representing word-context co-occurrences, we apply a technique to filter out contexts by relevance. The compressing technique consists in computing an *informativeness* measure between each word and their contexts (for instance, loglikelihood, mutual information, PMI, ...). Considering the experiments performed in Bordag (2008), we use loglikelihood as informativeness measure (Dunning, 1993). Then, for each word, only the R (relevant) contexts with highest loglikelihood scores are kept in the hash table. The top R contexts are considered to be the most *relevant* and informative for each word. R is a global, arbitrarily defined constant whose usual values range from 10 to 1000 (Biemann et al, 2013; Padró et al, 2014). However, this value can be computed by selecting a proportion over the total number of dependency contexts. In our work, $R = \sqrt{\|C\|}$, where $\|C\|$ is the total number of different contexts in the corpus. In short, we keep at most the R most relevant contexts for each target word.

A filtered model is then based on selecting the most relevant context per target word. It is an explicit representation. Methods based on dimensionality reduction and embeddings, by contrast, make the vector space more compact with dimensions that are not transparent in linguistic terms. The filtering-based approach turned out to be as efficient as other strategies based on dimensionality reduction such as SVD (Gamallo and Bordag, 2011).

4 Experiments and Evaluation

Four types of experiments were performed: rating by similarity, synonymy detection with multiple-choice questions, (near-)synonym detection using external thesaurus, and analogy tests.

The goal of these experiments is to compare our count-based word space with the embeddings described in Levy and Goldberg (2014a), which are publicly available[1] and which were generated using the `word2vec` software[2]. In addition, we will also compare the use of dependencies and bag-of-words. More precisely, four different models will be compared:

- A count-based model with syntactic dependencies, called *dep-count*.
- A predictive model with syntactic dependencies, called *dep-predict*.
- A count-based model with bag-of-words called *bow-count*.
- A predictive model with bag-of-words, called *bow-predict*.

The two predictive models, *dep-predict* and *bow-predict*, are based on the continuous *skip-gram* neural embedding model (Mikolov et al, 2013). The contexts of the two bag-of-words models, *bow-count* and *bow-predict*, were generated using a window of size 5, which is the number of words to the left and to the right of the target word.[3] The training algorithm is based on negative-sampling (without hierarchical softmax). The negative-sampling parameter (how many negative contexts to sample for every correct one) is 15. The subsampling method randomly removes words that are more frequent than some threshold t where $t = 10^{-5}$ in our experiments, according to the recommendation in Mikolov et al (2013).

The four models were built using the English Wikipedia (August 2013 dump) containing almost 2 billion tokens. To create the two dependency-based models, the corpus was parsed with a very specific configuration of the arc-eager transition-based dependency parser described in Goldberg and Nivre (2012). The performance of the parser for English is about 89% UAS (unlabeled attachment score) obtained on the CoNLL 2007 dataset.

To build the predictive models, target words appearing less than 100 times were filtered out. Likewise, contexts with frequency less than 100 were removed. The final model construction resulted in a vocabulary of about 175K target words for *dep-predict* and 183K for *bow-predict*. Concerning the contexts of the vector space, *dep-predict* is constituted by over 900K different dependency-based contexts and *bow-predict* by 183K context words. The two final embeddings contain 300 dimensions.

To build the count-based models, we also removed target words and contexts appearing less than 100 times. After having applied the process of context

[1] `https://levyomer.wordpress.com/2014/04/25/dependency-based-word-embeddings/`

[2] `code.google.com/p/word2vec/`

[3] We use *bow* to refer to linear bag-of-word contexts, which must be distinguished from CBOW (continuous bag-of-words). Unlike linear bag-of-words, CBOW uses continuous distributed representation of the context. It is a learning strategy that tries to predict a given word given its context, instead of predicting the context given a word as in the skip-gram model.

System	*Synonym*	*Topic*	*average*
dep-count	**0.7667**	0.5395	0.6531
dep-predict	**0.7580**	0.4920	0.625
bow-count	0.7148	**0.5929**	0.6538
bow-predict	0.7290	**0.6097**	0.6693

Table 1 Spearman correlation between the WordSim353 dataset and the rating obtained with the different systems.

filtering described in section 3.2, we obtained two hash tables: one with 203K target words and 435K different syntactic contexts (*dep-count* model), and another with 235K target words and 235K word contexts (*bow-count* model).[4] The distribution by categories of the target words in *dep-count* is the following: 165K nouns, 24K adjectives, and 14K verbs (similar distribution for the the rest of models). The R number of relevant syntactic contexts by word is $R = 659$ in *dep-count* and $R = 485$ in *bow-count*.

We use *Cosine* as similarity coefficient for all experiments and models.

4.1 Rating by Similarity

In the first experiments, we use the WordSim353 dataset (Finkelstein et al, 2002), which was constructed by asking humans to rate the degree of semantic similarity between two words on a numerical scale. Agirre et al (2009) split the dataset into two subsets: synonym/co-hyponym relations (tiger/cat) and topical relations (planet/astronomer). The performance of a computational system is measured in terms of correlation (Spearman) between the scores assigned by humans to the word pairs and the similarity Cosine coefficient assigned by the system taking into account the model space.

Table 1 summarizes the evaluation results. Concerning the synonym subset (first column), dependency-based approaches perform better than bow-based models. However, the differences between the two dependency-based approaches and the rest of strategies is statistically significant (Wilcoxon signed rank test, $p < 0.005$). In the topical subset (second column), the correlation scores of the dependency-based models drop dramatically. Semantic relatedness by topical similarity does not imply functional/paradigmatic relations, which are the basic relations extracted with syntactic based models. As it was expected, and also reported in Levy and Goldberg (2014a), count-based models perform significantly better than dependency-based in the topical subset, where semantic relations are beyond synonymy and co-hyponymy. In each column, the best groups of systems that do not differ significantly are emphasized (in bold).

The current state-of-the-art system for this dataset is a predictive model with bag-of-words, described in (Baroni et al, 2014a), which reaches 0.8 for

[4] The number of target words differs from predictive models due to multiple heuristics and thresholds (hyperparameters) used to generate both predictive and count-based models.

System	*Accuracy*
dep-count	**0.6458**
dep-predict	0.580
bow-predict	0.6041
bow-count	0.6041

Table 2 Accuracy of the evaluated systems on the ESL test (synonym detection)

System	*Noun Acc.*	*Adj. Acc.*	*Verb Acc.*	*Total Acc.*
dep-count	**0.8138**	0.7110	**0.7103**	**0.7552**
dep-predict	0.7558	0.6668	0.5982	0.682
bow-predict	0.7530	**0.7527**	0.6031	0.7049
bow-count	0.7230	0.6723	0.6181	0.6764

Table 3 Accuracy of the evaluated systems on the WBST test (synonym detection on nouns, adjectives, and verbs)

the synonym subset and 0.7 for the topical relations. That system was trained on a corpus of 2.8 billion tokens, which is larger than the corpus used in our experiments. It is worth noticing that, in the synonym subset, our best system (*dep-count-global*: 0.77 correlation) is close to the state-of-the-art.

4.2 Synonym Detection with Multiple-Choice Questions

In this evaluation task, a target word is presented with four synonym candidates, one of them being the correct synonym of the target. For instance, for the target *deserve*, the system must choose between *merit* (the correct one), *need*, *want*, and *expect*. Accuracy is the number of correct answers divided by the total number of words in the dataset. We used two datasets.

The fist dataset is ESL, constituted by 50 questions from the English as a Second Language test (Turney, 2001). Table 2 depicts the results, which are in fact quite far from the state-of-the-art: 0.86 accuracy reported in (Lu et al, 2011) using a count-based approach. The low performance is mainly due to the fact that the dataset contains several rare words, which were filtered out because they occur less than 100 times in the corpus. The *dep-count* model perform significantly better than the rest of systems ($p < 0.005$). However, the dataset is too small to infer relevant information.

The second dataset is an extended TOEFL test, called the WordNet-based Synonymy Test (WBST) proposed in (Freitag et al, 2005). WBST was produced by generating automatically a large set of TOEFL-like questions from the synonyms in WordNet. In total, this procedure yields 9,887 noun, 7,398 verb, and 5,824 adjective questions, a total of 23,509 questions. Table 3 shows the results. The *dep-count* model is the best system for nouns, verbs and for the whole dataset (last column). For all these sub-tests, the difference with regard to the second best model is statistically significant ($p < 0,005$). However, for adjectives the best system is *bow-predict*.

Words	*WordNet*	*OpenThesaurus*
nouns	21,414	13,681
verbs	4,873	3,761
adjectives	6,130	6,330
Total	32,417	23,772

Table 4 Target words of WordNet and OpenThesaurus also appearing in the evaluated models

Besides, most scores obtained by the best system are higher than those reported in Freitag et al (2005), the state-of-the-art for this task. More precisely, *dep-count* reaches 0.8138 accuracy for nouns, while the best system of the cited work achieves 0.758. With adjectives, the accuracy of our system is lower: 0.7110 *vs* 0.764; with verbs is higher: 0.762 *vs* 0.638; in total 0.7552 *vs* 0.722. More recently, an experiment on WBST dataset and reported in Zhu (2015), reached 0.69 accuracy using word embeddings and the Skip-Gram algorithm (Mikolov et al, 2013).

4.3 Thesaurus-Based Evaluation of Synonym/Hypernym Detection

4.3.1 Evaluation Protocol

One of the most large-scale evaluation protocols to measure the quality of synonymy (or near-synonymy) detection comprises as gold standard external lexical resources. It has been largely used for measuring the quality of count-based systems (Bordag, 2008). The gold standards we used in this task are WordNet (Fellbaum, 1998) and OpenThesaurus[5].

In our evaluation protocol, each model provides for each target word (common noun, adjective, or verb), a ranked list with its top-20 most similar words. Similarity was computed for words of these three categories separately. A similar word of the ranked list is considered as a true positive if it is *related* in the gold standard to the target word. In WordNet, we only consider synonymy (synsets) and hypernym relations (for only immediate hypernyms, i.e., direct ancestors). OpenThesaurus only provides relations between (near) synonyms. Target words to be evaluated are those that are found in the gold standard and in the five compared models. So, all models are compared against the same set of target words. Table 4 shows quantitative information in relation to the number of target (and then evaluated) words.

To measure the quality of the results provided by the five systems, we use Mean Average Precision (MAP) (Chen, 2003). Average Precision consists in evaluating the average quality of the ranking produced for each test word. More precisely, it is the average of the precision scores at the rank locations of each true positive. Mean Average Precision is the sum of average precisions

[5] `https://www.openthesaurus.de/`

System	*N1*	*N5*	*N20*	*V1*	*V5*	*V20*	*A1*	*A5*	*A20*
dep-count	**.1051**	**.0596**	**.0275**	**.2077**	**.1224**	**.0586**	**.1129**	**.0677**	**.0323**
dep-predict	.0720	.0377	.0165	.0959	.0628	.0280	.0714	.0407	.0184
bow-count	.0630	.0359	.0170	.1108	.0607	.0304	.0714	.0435	.0223
bow-predict	.0807	.0428	.0189	.1298	.0712	.0330	**.1122**	**.0652**	.0245

Table 5 MAP (top1, top5 and top20) obtained by the four systems against WordNet (only synonyms) for Nouns, Verbs, and Adjectives

System	*N1*	*N5*	*N20*	*V1*	*V5*	*V20*	*A1*	*A5*	*A20*
dep-count	**.1822**	**.1139**	**.0572**	**.3281**	**.1620**	**.1044**	**.1756**	**.1091**	**.0547**
dep-predict	.0720	.0377	.0165	.0959	.0628	.0280	.0967	.0631	.0184
bow-count	.1289	.0808	.0414	.1543	.0839	.0604	.1189	.0760	.0398
bow-predict	.1156	.0685	.0175	.1478	.0774	.0545	.1592	.0974	.0462

Table 6 MAP (top1, top5 and top20) obtained by the four systems against WordNet (synonyms and immediate hypernyms) for Nouns, Verbs, and Adjectives

System	*N1*	*N5*	*N20*	*V1*	*V5*	*V20*	*A1*	*A5*	*A20*
dep-count	**.1377**	**.0791**	**.0367**	**.2543**	**.1520**	**.0730**	**.1868**	**.1151**	**.0582**
dep-predict	.0720	.0377	.0165	.0959	.0628	.0280	.1071	.0574	.0270
bow-count	.0822	.0473	.0142	.1427	.0790	.0394	.1292	.0815	.0426
bow-predict	.1054	.0560	.0249	.1651	.0923	.0432	.1652	.1010	.0488

Table 7 MAP (top1, top5 and top20) obtained by the four systems against OpenThesaurus (only synonyms) for Nouns, Verbs, and Adjectives

divided by the number of evaluated words (i.e., words occurring in both the gold standard and the word model).

4.3.2 Results

Tables 5, 6 and 7 summarize the MAP scores obtained by the four systems, for the three syntactic categories at stake. In 5, we only consider as gold standard the synsets (synonymy) of WordNet. In 6, in addition to synonyms, we also consider hypernyms. For each category, several ranked lists from 1 to 20 words were taken into account. For instance, N20 means a ranked list containing the 20 most similar nouns to the target noun; V1 means the most similar verb to the target verb; A5 a list with the top 5 most similar adjectives. Even if the scores could seem to be quite low, they are in the same range as other similar experiments reported in previous work (Bordag, 2008; Gamallo, 2009). Notice also that the MAP score tends to go down as the ranked list grows up. Such a tendency is observed for all categories and systems. This is due to the fact that, for large ranked lists, the number of true positives might be much lower than the size of the ranked list.

The results show that, in most sub-tests, *dep-count* behaves significantly better than the rest of the systems ($p < 0.005$). The differences become more pronounced when the hypernyms are also considered for evaluation in addition to synonyms (see Table 6). It is worth noticing that even *bow-count* reaches better results than the predictive embeddings with hypernyms. As hypernyms

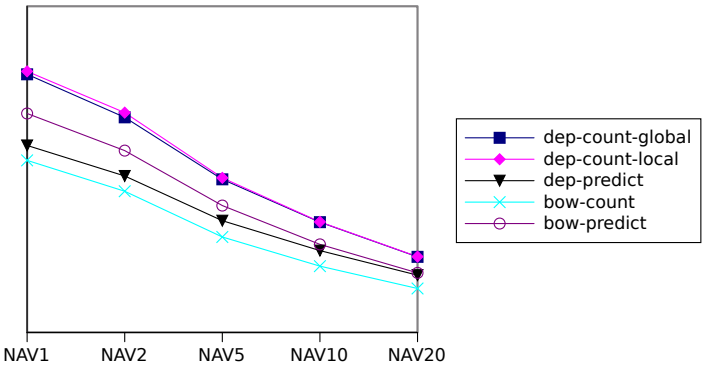

Fig. 1 MAP of the four systems by averaging from the three resources and the tree categories.

tend to be more generic and more frequent words than their hyponyms, it seems that count-based models boost frequent words while embeddings favor more uncommon words. This will be discussed later (Section 4.6).

The results also show that the differences between *dep-count* and the predictive models vary depending upon the syntactic category of the target word. Concerning the adjectives, there are no important differences. As Table 5 shows, the difference between *dep-count* and *bow-predict* is not statistically significant. However, with regard to nouns and verbs, the gap between *dep-count* and the rest of systems increases. This is in accordance with the previous task on the WBST dataset.

In sum, *dep-count* consistently outperforms predictive models by a non-trivial margin, as we can see more clearly in Figure 1. In this figure, all systems are compared by averaging over both the three gold standards and the three categories. Besides, it is worth noticing that *bow-predict* outperforms *dep-predict*. This is not in accordance with the experiments reported in (Levy and Goldberg, 2014b), where the dependency based embedding model reached better results than bag-of-word embeddings for the task of rating synonym/co-hyponym relations.

4.4 Analogy

So far, experiments have been performed to compare word models against their ability to encode *attributional* similarity between words. However, it is also possible to encode similarity between pairs of words, what is called *relational* similarity by Turney (2006). To compare the models, we used a dataset with analogy questions (Mikolov et al, 2013): given an example pair (*France, Paris*), and a test word (*Portugal*), the objective of the question is to find another word whose relation to the target word is the same as the relation illustrated

by the example pair (*Lisbon*). We then form the analogy "Paris is to France as Lisbon is to Portugal".

To compute relational similarity, we follow the two different algorithms defined in (Levy and Goldberg, 2014b). Given the analogy between two word pairs *a:b, c:d*:

$$\mathsf{argmax}_{d \in V} = sim(d, c) - sim(d, a) + sim(d, b) \quad (1)$$

$$\mathsf{argmax}_{d \in V} = \frac{sim(d, c) sim(d, a)}{sim(d, b) + \epsilon} \quad (2)$$

where V is the vocabulary excluding the question words, and *sim* stands for the similarity measure (Cosine). Besides, $\epsilon = 0.001$ is used to prevent division by zero.

In our experiments, we used the subset "capital-common-countries", containing 506 analogy questions from the Google test dataset reported in (Mikolov et al, 2013). Table 8 shows the results obtained by the five systems with the two algorithms: Adding corresponds to equation 1, while Mult. corresponds to equation 2. The following conclusions can be drawn from this experiment:

- As far as dependency based models are concerned, the multiplicative approach outperforms the addition one.
- In the case of bag-of-word based techniques, the addition approach clearly outperforms the multiplicative one. This is not in accordance with the results reported in (Levy and Goldberg, 2014b), where the multiplicative strategy reached the following scores: 0.905 accuracy by the embedding model and 0.994 by the count-based, using the same corpus as in our experiments (the English Wikipedia). The latter is the current state-of-the-art for this dataset.
- Dependency-based methods perform dramatically worse than bag-of-words models (both *bow-count* and *bow-predict*), which reach about 0.94 accuracy, close to the state-of-the-art.

It is worth noticing that the drop of performance by dependency-based models on the analogy task has been reported previously, but just in a footnote of the work by Levy and Goldberg (2014b). This observation could be confirmed with further experiments in future work, by enlarging the evaluation to the whole Google dataset. Such a bad performance might be explained as follows. The dependency-based models are suited to discover co-hyponyms and hyperonyms, which are often in paradigmatic relations. Paradigmatic relations exist between words outside the strings where they co-occur. Unlike syntagmatic relations, paradigmatic relations cannot be directly observed in utterances, that is why they are refered to as relations *in absentia*. By contrast, models based on bag-of-words may discover semantic relations between words co-occurring in the same sequence (syntagmatic relations). So, they tend to assign high similarity scores to word pairs related by other sematic relations than co-hyponymy or hyperonymy (e.g. capital-country relation). On the other hand,

System	*Adding*	*Mult.*
dep-count	0.2341	0.3310
dep-predict	0.3530	0.3786
bow-count	**0.9349**	**0.7475**
bow-predict	**0.9487**	**0.7491**

Table 8 Accuracy of the four systems on the analogy dataset "capital-common-countries", using addition and multiplicative strategies

the analogy task requires giving high similarity score to non-paradigmatic semantic relations. For instance, in the case of the capital-country relation, the analogy equation obtains high values when the system gives high scores to pairs such as "Paris" and "France", or "Madrid" and "Spain". In our experiments, the Cosine scores returned by the *bow-count* model for these two pairs are 0.59 and 0.62, respectively. However, the dependency models give low values to these pairs since they are neither co-hyponyms nor hyperonyms. The Cosine scores returned by our *dep-count* model for those two pairs are 0.07 and 0.08, respectively. This explains why the dependency-based models perform so badly in the analogy task.

4.5 Further Experiments

We also built an additional *dep-count* model using the output of a rule-based dependency parser, DepPattern (Gamallo and González, 2011; Gamallo, 2015). It means that the *dep-count* model was built twice: one version with DepPattern and another one with the transition-based dependency parser (Goldberg and Nivre, 2012) used in the reported experiments. All the experiments for *dep-count* were performed again, but now with the DepPattern model. As expected, the results between the two *dep-count* models were very similar: for all the tests we carried out (except the analogy task), the results we obtained were not statistically different according to the Wilcoxon signed rank test and for $p < 0.01$. However, the results using the transition-based dependency parser were slightly better. This is in accordance with the fact that the performance of this parser is also slightly better than DepPattern (about 89% against 83% UAS).

4.6 Discussion

Besides the performance and accuracy of the evaluated methods, we can also find relevant aspects and differences by analyzing the four systems from other viewpoints.

We analyzed the differences between the methods by comparing each pair of systems according to the proportion of common words over the total number of words returned by each system. This experiment was performed using the results obtained from the thesaurus-based evaluation task (Subsection 4.3).

Given two systems, we compare whether the top-10 most similar words returned by a system are also returned by the other one. Table 9 shows that the four algorithms yield quite different word models. The two most similar systems are *dep-predict* and *bow-predict*, with 29% common words. By contrast, the most dissimilar ones are *dep-predict* and *bow-count*: only 7%. More interestingly, we observe that the main differences are not due to the type of context (dependencies or bag-of-words), but to the type of vector space: embeddings or explicit counting. It follows that predictive and count-based models are complementary since they give very different outputs. By contrast, models based on bag-of-words and dependencies are less complementary because they tend to be right and to fail with the same examples, namely for the predictive approach. This is unfortunate because it means that their combination will produce poor gain in performance.

Another important difference between count-based and predictive models is that the latter favors words with lower frequencies. Table 10 shows the top 5 most similar words to "insane" and "veganism" returned by the two models. Frequencies are given in brackets. In the case of "insane", the most similar word returned by *dep-count* is "mad", which is a very frequent adjective also co-occurring with "insane" in a synset of WordNet. However, *dep-predict* penalizes the high frequency of "mad" and only ranks it as the 39th most similar word to "insane". Likewise, in the case of "veganism", *dep-predict* penalizes the noun "vegetarianism" (3rd instead of 1st as in *dep-count*) probably because its frequency is four times higher than the target word.

In order to interpret such a difference, we hypothesize that predictive models take some risks since only produce *approximations* to the actual corpus distributions, which may improve representations, but also skew them. By contrast, our count-based technique is a more conservative strategy because it takes decisions based on a great amount of observed word-context co-occurrences. As predictive models are able to learn from few observed data, rare and low-frequency words may appear very often at the upper ranks in the similarity task. However, count-based models tend to give more reliability to those words observed in the corpus in several contexts; so frequent words are more likely to appear at the upper similarity ranks.

Nevertheless, the differences between count-based and predictive models could be due to external parameters that are not at the core of the algorithms used to build the word models. As Levy et al (2015) suggest, much of the difference between vectorial models are due to certain system design choices and hyperparameter optimizations (e.g., subsampling frequent words, window size, etc.) rather than the algorithms themselves. The authors revealed that seemingly minor variations in external parameters can have a large impact on the success of word representation methods.

	dep-count	bow-count	dep-predict	bow-predict
dep-count	100	20	15	14
bow-count		100	7	16
dep-predict			100	29
bow-predict				100

Table 9 Proportion (%) of common words over the total number of returned words between pairs of methods

word	*dep-count*	*dep-predict*
insane (12443)	mad (36778), psychotic (3553), evil (70718), paranoid (4504), cruel (9258)	senile (986), impotent (932), catatonic (828), repentant (565), jobless (766)
veganism (361)	vegetarianism (1379), pacifism (1508), vegan (2918), weight-loss (424), relativism (1227)	transhumanism (314), nudism (249), vegetarianism (1379), ageism (229), scientism (237)

Table 10 The top 5 most similar words of "insane" and "veganism" given by two models (frequencies in brackets)

5 Conclusions

The experiments that we have reported in this article provide elements to confirm that useful lexical information can be extracted from simple co-occurrence statistics (count-based and explicit word vectors) using straightforward similarity metrics.

Word representations based on explicit co-occurrences between words and syntactic contexts can be stored in an efficient way using filtering strategies on hash tables. Besides, they can outperform neural embeddings of word/syntactic-contexts in just those tasks where syntactic contexts behave better than bag-of-words, namely synonym and hypernym detection.

Other relevant but preliminary conclusions that can be drawn from the reported experiments are the following. With regard to synonymy-hypernym detection task:

- Count-based models with syntactic contexts outperforms predictive models with syntactic contexts. To the best of our knowledge, this experiment had not been previously carried out and, therefore, can be considered as the main contribution of our work.
- Count-based models with syntactic contexts perform significantly better than count-based models with *bow* contexts. This observation has largely been reported in previous work.
- Predictive models with *bow* contexts perform (in most experiments) significantly better than predictive models with syntactic contexts (however, in the experiment described in Section 4.1 the dependency based method *dep-predict* outperforms *bow-predict*).
- Predictive models with *bow* contexts perform in some experiments significantly better than count-based models with *bow* contexts.

With regard to the analogy test, *bow* models (both predictive and count based) clearly outperform syntactic models.

Obviously, these conclusions must be relativized, as they were drawn from experiments restricted to the English language and using large well parsed corpora. They cannot be extrapolated to other languages than English and to other linguistic tasks.

However, we may summarize that predictive word models built with (shallow) neural learning strategies are not always able to improve over traditional word spaces that the NLP community has been designing and used for about two decades.

The two count-based models described in the article are publicly available.[6]

Acknowledgments

This research has been partially funded by the Spanish Ministry of Economy and Competitiveness through project FFI2014-51978-C2-1-R. We are very grateful to Omer Levy and Yoav Goldberg for sending us the parsed corpus used to build their embeddings. Moreover, we are also very grateful to the reviewers for their useful comments and suggestions.

References

Agirre E, Alfonseca E, Hall K, Kravalova J, Paşca M, Soroa A (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, NAACL '09, pp 19–27

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

Baroni M, Bernardi R, Zamparelli R (2014a) Frege in space: A program for compositional distributional semantics. LiLT 9:241–346

Baroni M, Dinu G, Kruszewski G (2014b) 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), Baltimore, Maryland, pp 238–247

Biemann, C, Riedl, M (2013) Text: Now in 2d! a framework for lexical expansion with contextual similarity. Journal of Language Modelling 1(1):55–95

Blacoe W, Lapata M (2012) A comparison of vector-based representations for semantic composition. In: Empirical Methods in Natural Language Processing - EMNLP-2012, Jeju Island, Korea, pp 546–556

Bordag S (2008) A Comparison of Co-occurrence and Similarity Measures as Simulations of Context. In: 9th CICLing, pp 52–63

[6] `http://fegalaz.usc.es/~gamallo/resources/count-models.tar.gz`

Bullinaria JA, Levy JP (2007) Extracting semantic representations from word co-occurrence statistics: A computational study. Behavior Research Methods pp 510–526

Bullinaria JA, Levy JP (2013) Limiting factors for mapping corpus-based semantic representations to brain activity. PLoS ONE 8(3)

Chen Z (2003) Assessing sequence comparison methods with the average precision criterion. Bioinformatics 19

Collobert R, Weston J (2008) A unified architecture for natural language processing: Deep neural networks with multitask learning. In: International Conference on Machine Learning, ICML

Curran JR, Moens M (2002) Improvements in Automatic Thesaurus Extraction. In: ACL Workshop on Unsupervised Lexical Acquisition, Philadelphia, pp 59–66

Dunning T (1993) Accurate methods for the statistics of surprise and coincidence. Computational Linguistics 19(1):61–74

Fellbaum C (1998) A semantic network of English: The mother of all WordNets. Computer and the Humanities 32:209–220

Finkelstein L, Gabrilovich E, Matias Y, Rivlin E, Solan Z, Wolfman G, Ruppin E (2002) Placing search in context: the concept revisited. ACM Trans Inf Syst 20(1):116–131

Freitag D, Blume M, Byrnes J, Chow E, Kapadia S, Rohwer R, Wang Z (2005) New experiments in distributional representations of synonymy. In: Proceedings of the Ninth Conference on Computational Natural Language Learning, pp 25–32

Gamallo P (2008) Comparing window and syntax based strategies for semantic extraction. In: PROPOR-2008, Lecture Notes in Computer Science, Springer-Verlag, pp 41–50

Gamallo P (2009) Comparing different properties involved in word similarity extraction. In: 14th Portuguese Conference on Artificial Intelligence (EPIA'09), LNCS, Vol. 5816, Springer-Verlag, Aveiro, Portugal, pp 634–645

Gamallo P (2015) Dependency parsing with compression rules. In: International Workshop on Parsing Technology (IWPT 2015), Bilbao, Spain

Gamallo P, Bordag S (2011) Is singular value decomposition useful for word simalirity extraction. Language Resources and Evaluation 45(2):95–119

Gamallo P, González I (2011) A grammatical formalism based on patterns of part-of-speech tags. International Journal of Corpus Linguistics 16(1):45–71

Gamallo P, Agustini A, Lopes G (2005) Clustering Syntactic Positions with Similar Semantic Requirements. Computational Linguistics 31(1):107–146

Goldberg Y, Nivre J (2012) A dynamic oracle for arc-eager dependency parsing. In: COLING 2012, 24th International Conference on Computational Linguistics Proceedings of the Conference: Technical Papers, 8-15, Mumbai, India, pp 959–976

Grefenstette G (1993) Evaluation techniques for automatic semantic extraction: Comparing syntactic and window-based approaches. In: Workshop on Acquisition of Lexical Knowledge from Text SIGLEX/ACL, Columbus, OH

Harris Z (1985) Distributional Structure. In: Katz J (ed) The Philosophy of Linguistics, New York: Oxford University Press, pp 26–47

Hofmann M, Kuchinke L, Biemann C, Tamm S, Jacobs A (2011) Remembering words in context as predicted by an associative read-out model. Frontiers in Psychology 252(2):85–104

Hofmann MJ, Jacobs AM (2014) Interactive activation and competition models and semantic context: From behavioral to brain data. Neuroscience and Biobehavioral Reviews 46(Part 1):85–104

Huang E, Socher R, Manning C (2012) Improving word representations via global context and multiple word prototypes. In: ACL-2012, Jeju Island, Korea, pp 873–882

Landauer T, Dumais S (1997) A solution to Plato's problem: The Latent Semantic Analysis theory of acquision, induction and representation of knowledge. Psychological Review 10(2):211–240

Lebret R, Collobert R (2015) Rehabilitation of count-based models for word vector representations. In: Gelbukh AF (ed) CICLing (1), Springer, Lecture Notes in Computer Science, vol 9041, pp 417–429

Levy O, Goldberg Y (2014a) Dependency-based word embeddings. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, pp 302–308

Levy O, Goldberg Y (2014b) Linguistic regularities in sparse and explicit word representations. In: Proceedings of the Eighteenth Conference on Computational Natural Language Learning, CoNLL 2014, Baltimore, Maryland, USA, June 26-27, 2014, pp 171–180

Levy O, Goldberg Y (2014c) Neural word embedding as implicit matrix factorization. In: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, pp 2177–2185

Levy O, Goldberg Y, Dagan I (2015) Improving distributional similarity with lessons learned from word embeddings. Transactions of the Association for Computational Linguistics 3:211–225

Lin D (1998) Automatic Retrieval and Clustering of Similar Words. In: COLING-ACL'98, Montreal

Lu CH, Ong CS, Hsub WL, Leeb HK (2011) Using filtered second order co-occurrence matrix to improve the traditional co-occurrence model. In: Computer Technologies and Information Sciences, Department of Computer Science and Information Engineering, National Taiwan University, URL `http://www.osti.gov/eprints/topicpages/documents/record/803/2113132.html`

Mikolov T, Yih Wt, Zweig G (2013) Linguistic regularities in continuous space word representations. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta, Georgia, pp 746–751

Padó S, Lapata M (2007) Dependency-Based Construction of Semantic Space Models. Computational Linguistics 33(2):161–199

Padró M, Idiart M, Villavicencio A, Ramisch C (2014) Nothing like good old frequency: Studying context filters for distributional thesauri. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pp 419–424

Peirsman Y, Heylen K, Speelman D (2007) Finding semantically related words in Dutch. Co-occurrences versus syntactic contexts. In: CoSMO Workshop, Roskilde, Denmark, pp 9–16

Seretan V, Wehrli E (2006) Accurate Collocation Extraction Using a Multilingual Parser. In: 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL, pp 953–960

Turney P (2001) Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In: 12th European Conference of Machine Learning, pp 491–502

Turney PD (2006) Similarity of semantic relations. Comput Linguist 32(3):379–416

Zhu P (2015) N-grams based linguistic search engine. International Journal of Computational Linguistics Research 6(1)