Single or Multiple? Combining Word Representations Independently Learned from Text and WordNet

Josu Goikoetxea, Eneko Agirre, and Aitor Soroa
IXA NLP Group
University of the Basque Country
Donostia, Basque Country
{josu.goikoetxea,e.agirre,a.soroa}@ehu.eus

Abstract

Text and Knowledge Bases are complementary sources of information. Given the success of distributed word representations learned from text, several techniques to infuse additional information from sources like WordNet into word representations have been proposed. In this paper, we follow an alternative route. We learn word representations from text and WordNet independently, and then explore simple and sophisticated methods to combine them. The combined representations are applied to an extensive set of datasets on word similarity and relatedness. Simple combination methods happen to perform better that more complex methods like CCA or retrofitting, showing that, in the case of WordNet, learning word representations separately is preferable to learning one single representation space or adding WordNet information directly. A key factor, which we illustrate with examples, is that the WordNet-based representations captures similarity relations encoded in WordNet better than retrofitting.

In addition, we show that the average of the similarities from six word representations yields results beyond the state-of-the-art in several datasets, reinforcing the opportunities to explore further combination techniques.

Introduction

Word embeddings successfully capture lexical semantic information about words based on co-occurrence patterns extracted from large corpora, with excellent results on several tasks, including word similarity (Collobert and Weston 2008; Turian, Ratinov, and Bengio 2010; Socher et al. 2011). More recently, the combination of distributed word representations and knowledge bases (KBs) is gaining strength, exploring the use of KBs such as WordNet (Miller 1995), FreeBase (Bollacker et al. 2008) and PPDB (Ganitkevitch, Van Durme, and Callison-Burch 2013) on tasks such as word similarity and synonym selection (Faruqui et al. 2015), or analogy (Rastogi, Van Durme, and Arora 2015). These methods are built on text-based methods (Mikolov, Yih, and Zweig 2013; Pennington, Socher, and Manning 2014; Huang et al. 2012) and use flat lists of relations from KBs to either modify the learning algorithms (Halawi et al. 2012; Wang et al. 2014a; Tian et al. 2015; Rastogi, Van Durme, and Arora 2015) or post-process previously learned word representations (Faruqui et al. 2015).

This paper departs from previous work in that it explores the combination of word representations which have been independently learned from different sources like text corpora or WordNet. We tried several simple combination techniques like, among others, averaging similarity results or concatenating vectors, and more complex methods like canonical correlation analysis (Faruqui and Dyer 2014) or the recently proposed retrofitting (Faruqui et al. 2015). In particular, we combine state-of-the-art corpus-based word representations (Mikolov et al. 2013a) and WordNet-based word representations (Goikoetxea, Soroa, and Agirre 2015). The simple methods outperform the more complex methods, as shown in several similarity and relatedness datasets. In the case of retrofitting, we show that our WordNet-based embeddings are able to represent the rich information in WordNet.

The paper is structured as follows. We first present related work, followed by the methods to produce our word representations. The next section presents several combination techniques, followed by the experiments combining two representations. We then present additional word representations and their combinations.

Related Work

Recently, several researchers have tried to infuse knowledge from KBs into corpus-based word representations. (Halawi et al. 2012) incorporated KB information in low-dimensional word embeddings adding WordNet-based constraints to the optimization process. The constraints are given by a set of related word pairs in WordNet, which need to be close in the embedding space.

(Wang et al. 2014b) introduced typed relations between entities from Freebase in the learning algorithm, projecting entity embeddings onto an hyperplane w.r.t. the KB relation and using those projected vectors in the scoring function. (Tian et al. 2015) improved the method by adopting different projections for head and tail entities when defining the loss function.

(Faruqui and Dyer 2014) find the correlations between two sets of multidimensional variables in order to combine word spaces coming from two different languages. They projected their original monolingual word vectors to a shared (multilingual) space, improving their performance in

similarity and analogy tasks. We will apply their technique to combine two spaces in the same language.

(Faruqui et al. 2015) proposed a graph-based method to incorporate relational information from KBs. Contrary to previous techniques they can improve the quality of any pre-existing word representations. The algorithm optimizes euclidean distances between words which are related in the KB. They applied it to pre-existing vectors like Skip-gram (Mikolov et al. 2013b) using relational information from WordNet, PPDB (Ganitkevitch, Van Durme, and Callison-Burch 2013) and FrameNet (Baker, Fillmore, and Lowe 1998). We will compare retrofitting head to head to our combination methods in the "Comparison to retrofitting" section.

In related work, (Rastogi, Van Durme, and Arora 2015) generalized the traditional LSA single-view model by the Generalized Canonical Correlation Analysis, so that the model is able to introduce different sources of information when learning. They combined the English part of the polyglot Wikipedia (Al-Rfou, Perozzi, and Skiena 2013), a large bitext corpus from PPDB project and the Annotated Gigaword Corpus (Napoles, Gormley, and Van Durme 2012), as well as relational information from WordNet, FrameNet, PPDB, CatVar (Habash and Dorr 2003) and morpha (Minnen, Carroll, and Pearce 2001). They evaluated their multi-view approach in various similarity and relatedness datasets, achieving competitive results with the state of the art.

Following a different strategy, in (Goikoetxea, Soroa, and Agirre 2015) we encoded the structure of WordNet, combining a random walk algorithm and dimensionality reduction. We created a pseudo-corpus with the words emitted in a random walk over WordNet, and then produced a distributed word representation using word2vec[1]. Contrary to the previous proposals, our method only uses information in WordNet. We will build on this work in our experiments.

Word Representations

In our experiments we learn word representations from information coming from textual corpora and KBs. When needed, we tuned free parameters in one of the smallest similarity datasets (RG, see the "Datasets and Evaluation" section).

Text-based representations

Following (Baroni, Dinu, and Kruszewski 2014), we created a corpus, which we call WBU, by concatenating the English Wikipedia[2], the British National Corpus[3] and ukWaC[4]. The corpus comprises $5 \cdot 10^9$ tokens. Using this corpus, we applied a skipgram Neural Network Language Model (NNLM) (Mikolov et al. 2013a)[5] in order to produce the representations (**WBU** for short). In the skipgram model each current word is input to a log-linear classifier with a continuous projection layer, which predicts the previous and subsequent words in a context window. We optimized the parameters on RG, resulting in the following: dimensionality of 300, 5 negative samples, sub-sampling threshold of zero and window size of 5 words.

WordNet-based representations

We used the method described in (Goikoetxea, Soroa, and Agirre 2015), which combines random walks over KBs and NNLMs in order to produce the word representations.

Given the novelty of these representations, we explain them in more detail.

The random walks generate compact contexts (a so-called pseudo-corpus) that implicitly contain the graph's semantic relations and thus encode the structure of the KB. The graph is formalized as $G = (V, E)$, where V is the set of concepts and E the undirected links between concepts. A dictionary encodes the lexicalization of concepts as words. The algorithm uses as inputs the graph, the dictionary and a restart probability α. Firstly it selects the starting point of the random walk, which is going to be a random vertex from V. Then, at every step of the random walk, the algorithm tosses a coin to restart the random walk (with probability $1-\alpha$), or to select a neighbor vertex at random and continue the random walk. In each vertex, it emits one of the lexicalizations of the concept at random. When the random walk restarts, the emitted words are included in the pseudo-corpus as a sentence, and a new starting vertex is selected. The walk halts after a pre-defined number of restarts.

In our experiments we derived the graph and dictionary from WordNet 3.0 with gloss relations[6]. The graph comprises 117.522 vertices (synsets) and 525.356 edges (semantic relations). We ran the random walk for $2 \cdot 10^8$ restarts, emitting $1.1 \cdot 10^9$ tokens. The pseudo-corpus was fed into the same skipgram model as for text-based representations, producing dense WordNet-based representations (**RWwn** for short) which encode the structural information in WordNet. We used the parameters reported in (Goikoetxea, Soroa, and Agirre 2015)[7], but we run more iterations of the random walk algorithm, as it produced slightly better results on RG. Note that, contrary to text corpora, the amount of effective information is delimited by both the graph and the number of random walks, but further iterations did not improve results in RG.

The random walk produces contexts like the following: *yucatec mayan quiche kekchi speak sino-tibetan tone_language west_chadic talk*.The example shows how the contexts encode implicitly the information in WordNet. It starts with *yucatec*, which is a mayan language, followed by *mayan*, followed by two different spellings of another mayan language *quiche* and *kekchi*, followed by a related verb *speak*, and goes on with some other different languages or language-related terms. Contexts may contain multiwords, nouns, verbs, adjectives and adverbs.

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

[2] linguatools.org/tools/corpora/wikipedia-monolingual-corpora/

[3] http://www.natcorp.ox.ac.uk

[4] http://wacky.sslmit.unibo.it

[5] https://code.google.com/p/word2vec/

[6] http://wordnet.princeton.edu/glosstag.shtml

[7] Dimensionality of 300, 3 iterations, 5 negative samples, window size 5.

Combination methods

We tried several combinations of word representations. When needed, we tuned free parameters using RG as the development dataset.

Simple vector combinations

Two vectorial word representations can be easily combined concatenating the two vectors (CAT), computing the centroid (CEN), or creating a complex number (CMP). The latter, is based on the proposal of (Wittek et al. 2014) in which, on the one hand, they used the real component of the complex vectors to represent the distributional semantics, and on the other, they encoded the ontological data in the imaginary component. In our particular case, we introduced the corpora-based embeddings in the real part and the WordNet-based ones in the imaginary part.

These combinations cannot be applied to vectors of different dimensionality. In the case of concatenation, although technically possible, the vector with higher dimensionality will tend to dominate the combined similarity results.

Correlation analysis

Principal Component Analysis (PCA) (Jolliffe 2014) is a technique to reduce dimensionality, where the original variables are replaced by a smaller number of derived variables, the principal components (linear combinations of the original variables). We applied PCA to the concatenated representations of all target tokens in the datasets (cf. "Datasets and Evaluation" section)[8]. We tuned the dimensionality of the PCA word-vectors in RG, yielding the best result with 300 dimensions. PCA can be seen as the lower-dimensionality representation of CAT.

Canonical Correlation Analysis (CCA) also explores linear correlations, but in this case between two sets of multidimensional variables. As mentioned in the related work, CCA has been already used to combine different word representations (Faruqui and Dyer 2014), who used it to incorporate evidence from two language corpora.

In our case, the two representations have overlapping vocabulary, and thus we apply CCA[9] to maximize the correlation for the two vectors of the same token in the projected space. We tuned the dimensionality of the projected space and the two projection possibilities (project the text representations on the shared space, or the WordNet-based representations on the shared space) on the RG dataset, achieving the best results for 180 dimensions when projecting the WordNet-based representations.

Corpus combination

Given that both our text-based representation and WordNet-based representations use corpora (text corpora in one case, automatically generated pseudo-corpus in the second case), we can shuffle the two corpus and then use the distributional vector to obtain a single representation (COR). We combined both corpora, without tuning any parameter.

[8]http://finzi.psych.upenn.edu/R/library/mdatools/html/pca.html

[9]http://cs.cmu.edu/~mfaruqui/soft.html

Result combination

This method combines the similarity scores for each word representations. Given the cosine similarity values of the two representations, the combined similarity will be the average of the two values (AVG). In another variant, we first order the pairs in the dataset according to the similarity values in ascending order, and then average the ranks (RNK). The motivation for combining ranks is that Spearman is the main evaluation measure used in word similarity, and Spearman disregards the actual values and computes the correlation based on the relative rank of the values. Note that, in these cases,the representations are not combined.

Experiments

In order to evaluate word representations, we decided to follow the large body of literature on word similarity and relatedness (Agirre et al. 2009). The similarity of two words was calculated as the cosine between the respective word representations, except for the complex number CMP combination, where we use the complex vector cosine similarity (Scharnhorst 2001).

Datasets and Evaluation

We have used several popular datasets for evaluation. The first three are similarity datasets: RG (Rubenstein and Goodenough 1965) and SimLex999 (SL) (Hill, Reichart, and Korhonen 2014) and WordSim353 Similarity (WSS) (Agirre et al. 2009). The other four are relatedness datasets: WordSim353 Relatedness (WSR) (Agirre et al. 2009), MTURK287 (MTU) (Radinsky et al. 2011), MEN (Bruni, Tran, and Baroni 2014) and the full WordSim353 (WS) (Gabrilovich and Markovitch 2007). In addition, we report the average for the similarity datasets (RG, SL, WSS) and the relatedness datasets (WSR, MTU, MEN). Given the fact that WSS and WSR are subsets of WS, we use WSR when averaging relatedness, WSS for similarity and the full WS when averaging all. The evaluation measure computes the rank correlation (Spearman) between human judgments and system values, as customary. We performed significance tests using Fisher's z-transformation (Press et al. 2002, equation 14.5.10).

Combining text-based and WordNet-based embeddings

In our first experiment, Table 1 reports the results of our WordNet-based representations (RWwn, first row) with text-based embeddings (WBU, second row) and the combinations. The results show that both methods perform a la par for relatedness-based datasets, and that WordNet enables better results in similarity.

More importantly, the table reports the gain with respect to WBU for each combination technique. CAT and PCA over the concatenated representations outperform all other techniques on similarity, relatedness and overall, followed by AVG and COR. The rest perform lower, with RNK and CMP underperforming the text-based embeddings. Special mention goes to CCA, which, contrary to the work on multiple language embeddings (Faruqui and Dyer 2014), shows

	RG	SL	WSS	WSR	MTU	MEN	WS	sim	rel	all
RWwn	82.3	52.5	76.2	58.7	62.1	75.4	68.7	70.3	65.4	68.2
WBU	76.4	39.7	76.6	61.5	64.6	74.6	67.3	64.2	66.9	64.5
CAT	7.8	**12.5**	6.7	6.5	7.5	**6.0**	8.0	9.0	**6.7**	8.4
CEN	4.6	9.6	2.7	-1.1	1.3	3.2	2.3	5.6	1.2	4.2
CMP	-3.4	-1.2	-2.9	-8.9	-7.4	-0.9	-6.9	-2.5	-5.7	-4.0
PCA	**10.8**	**12.5**	5.7	5.3	**8.3**	5.6	6.9	**9.6**	6.5	**8.9**
CCA	6,8	2,7	-0,4	-0,2	11,7	-6,1	-3,5	6,0	-3,3	2,3
COR	6.6	8.2	**7.2**	8.8	3.3	4.1	**8.6**	7.4	5.4	6.2
AVG	8.0	12.1	5.5	6.5	7.0	6.2	7.4	8.5	6.6	8.2
RNK	7.3	11.3	0.2	**11.7**	-14.7	-14.7	6.6	6.2	-5.9	-0.8

Table 1: Results as Spearman for WordNet-based representations (RWwn, firs row), skipgram on text corpora (WBU, second row), and the **absolute gain with respect to WBU** when combined with WordNet-based representations (rest of rows). Leftmost columns for single datasets, and rightmost for averages across similarity, relatedness and all datasets. Best results in each column in bold. Regarding the "all" column, PCA, CAT and AVG are significantly better than the rest (99% confidence level), but not among themselves (even at 95% confidence level).

	RG	SL	WSS	WSR	MTU	MEN	WS	sim	rel	all
FAR	74.8	43.7	74.1	61.0	69.9	68.0	65.6	64.2	66.5	64.4
+WN$_{sh}$	5.0	7.4	4.0	-1.1	-0.6	2.6	1.9	5.5	0.3	3.3
+WN$_{all}$	4.9	2.5	2.6	4.3	2.4	5.7	3.7	3.3	4.1	3.9
WBU	76.4	39.7	76.6	61.5	64.6	74.6	67.3	64.2	66.9	64.5
+WN$_{sh}$	4,6	-12,2	-4,8	-18,6	8,0	-4,9	-2,7	2,6	-4,3	1,3
+WN$_{all}$	6,3	0,9	2,3	0,2	2,4	0,9	0,9	3,7	0,3	2,1
PCA	**10.8**	**12.5**	**5.7**	**5.3**	**8.3**	**5.6**	**6.9**	**9.6**	**6.5**	**8.9**

Table 2: Results as Spearman for the embeddings used in (Faruqui and Dyer 2014) (FAR, top row), and the absolute gain when using retrofitting to combine two varieties of WordNet information (+WN$_{sh}$ and +WN$_{all}$). We also include the results for skipgram on text corpora (WBU), and the respective retrofitting gains. We show our combined results again (PCA) for easier comparison. Best results in each column in bold. Regarding the "all" column, FAR+WN$_{all}$ and FAR+WN$_{sh}$ are significantly better than FAR (99% confidence level), but WBU+WN$_{all}$ and WBU+WN$_{sh}$ are not significantly better than WBU (even at 95% confidence level), which shows that retrofitting is only able to weakly profit from WordNet information. On the other hand, PCA is significantly better than all other results (99% confidence level).

a weak improvement, even if the larger improvements with CAT show that the two representations are complementary.

PCA slightly outperforms CAT with just 300 dimensions instead of 600, showing that some of the dimensions are linearly correlated, at least for these datasets. It also shows that it is possible to produce a compact, 300-dimension space which is much more effective than the original text-based and WordNet-based representations, which also have 300 dimensions.

Comparison to retrofitting

The comparison to retrofitting (Faruqui et al. 2015) deserves some more attention. Retrofitting included experiments infusing knowledge from WordNet, among others. The method converts the knowledge into a flat list of related words, that is, for each word in the vocabulary, a list of similar or closely related words is given as input to the algorithm. The algorithm then tweaks the original vectors of related words so they are close to each other. In their work, they extracted synonymy and hypernymy relations from WordNet (WN$_{sh}$), getting better results when using both relations. As our WordNet-based method uses additional relations, including gloss relations from WordNet, we produced an additional list of closely related words (WN$_{all}$)[10], where, for each target word, we list all words with a direct relation in WordNet.

The top three rows in Table 2 shows that retrofitting all relations in WordNet (+WN$_{all}$) to the embeddings used in (Faruqui et al. 2015)[11] produces slightly better results than using synonymy and hypernymy alone (+WN$_{sh}$), with worse results on similarity and better results on relatedness. This makes sense, as synonymy and hypernymy are linked to similarity, and gloss relations to relatedness (Agirre et al. 2009).

The other rows in Table 2 show the gains of retrofitting when applied on our WBU embeddings. The gains here are smaller, perhaps because of the different techniques and parameters used to produce the text-based embeddings[12]. In any case, both synonyms and hypernyms (+WN$_{sh}$) and all relations (+WN$_{all}$) produce improvements, with slightly higher results when using all relations. Still, the gain is small compared to the gains obtained when combining our representations, which obtains the best results across all datasets.

The higher results of our combination can be due to the fact that the WordNet-based embeddings are able to capture better the nuances of words and senses, as well as the fact that random walks capture relations and similarities of words beyond direct relations. Retrofitting, in contrast, models only direct relations between words.

For instance, the word pair *physics-proton* is given a high relatedness value in WS (8.12 out of 10), being the 45th most related pair in the dataset (rank 45), with pair *tiger-tiger* having 10 and being the the most related pair (rank 1)[13].

[10]Note that in the retrofitting paper, the authors refer to the use of synonyms and hypernyms as WN$_{all}$, while in this paper we do use all relations in WordNet, including part-of, gloss relations, etc. Their WN$_{all}$ is thus our WN$_{sh}$.

[11]The authors use the word2vec embeddings available in https://code.google.com/p/word2vec/

[12]Similar variability is reported by the authors of the retrofitting paper.

[13]We will refer to the rank in the dataset in order to compare the similarity values returned for each representations, as the absolute values might not be directly comparable. In addition, the evaluation

	RG	SL	WSS	WSR	MTU	MEN	WS
txt	—	—	—	—	69.2	—	74.4
CLEAR gain	—	—	—	—	-0.5	—	2.3
txt	71.2	34.5	76.8	60.1	59.1	71.4	68.0
MVLSA gain	9.6	9.4	2.4	3.4	3.8	4.4	2.1
txt	—	—	—	—	—	—	64.7
FREEBASE gain	—	—	—	—	—	—	3.7

Table 3: Results as Spearman for additional work which enriches text-based embeddings. For each technique, we report the text-only embeddings (txt) and the gain when adding other information sources, see text for more details.

There is no direct relation between this pair in WordNet, but the WordNet-based embeddings assign it a large value (rank 41), with the CAT combination also ranking it high (rank 42), even if the text-based WBU would rank them low (rank 170). Retrofitting cannot make this pair any closer, as it has no access to the fact that they are indirectly related in WordNet. In fact, retrofitting makes this pair rank even lower than WBU (rank 193), perhaps because other relations make move them apart.

On the negative side, retrofitting is able to include simple information like that from PPDB, which just lists paraphrase probabilities for word and phrase pairs. In fact, (Faruqui et al. 2015) reported improvements when retrofitting with PPDB. In our case, it is necessary to first produce full-fledged embeddings for each resource (e.g. PPDB) and then combine. We tried to do that using PPDB, but failed to produce meaningful representations.

Comparison to other combinations

Table 3 shows the results for other techniques that add knowledge from external resources into text embeddings (cf. related work section). We already reported on retrofitting, and we thus focus on CLEAR (Halawi et al. 2012) and Multiview LSA (Rastogi, Van Durme, and Arora 2015) (MVLSA for short). CLEAR used as a baseline the results yielded by training their own text-based distributional model with the Yahoo! Answers corpus[14] in 100-dimensional space. We chose the best reported results when applying WordNet restrictions, in the form of pairs of synonyms, hypernyms, and meronyms.

(Rastogi, Van Durme, and Arora 2015) trained the English part of the Polyglot Wikipedia dataset released by (Al-Rfou, Perozzi, and Skiena 2013) with their LSA-based model multiview model using 300 dimensions. They used WordNet in addition to other information sources.

In these cases, the results between systems are not directly comparable, as the different research works use both a different baseline text-based representation and a different subset of WordNet relations. Still, the fact that we obtain the largest absolute gain in all datasets is an indication that our alternative approach is worth following.

In a different strand of work, (Tian et al. 2015) infuses information from relations in Freebase into text-based embeddings (FREEB). Again, an approximate comparison can be made. The information used is somehow comparable to WBU, with Freebase relations closely related to the relations in Wikipedia (which we used, in addition to hyperlinks, to produce Wikipedia embeddings, as described in the "Combining more than two sources" section). For comparison's sake, we combined WBU and these embeddings, obtaining a gain of 4.6 (on top of the 64.5 for WBU), exhibiting higher gain and higher total performance.

Combining more than two sources

Given the good results when combining two word representations, we explored the combination of several other word representations.

Additional word representations

Regarding text-based embeddings, in addition to WBU word representations, we downloaded the representations which were released by the word2vec authors[15] (GOOG for short). These vectors had been trained on the Google News corpora, about $100 \cdot 10^9$ tokens, using a different but related NNLM called CBOW, with the following parameters: vector size 300, 3 negative samples, sub-sampling threshold of 10^{-5} and window size of 5 words. These vectors have been built using a larger corpus, but the parameters have not been optimized on word similarity tasks. The fact that they use a different corpus make them complementary to WBU vectors.

Regarding random walk based embeddings, we decided to apply them to the Wikipedia graph described in (Agirre, Barrena, and Soroa 2015) which is publicly available. V consists of all Wikipedia pages except redirect, disambiguation and category pages. The Wikipedia graph includes a edges between pages $a1$ and $a2$ if and only if there exists a link from $a1$ to $a2$ and from $a2$ to $a1$. This KB is formed with 2.955.154 nodes (articles) and 16.338.664 edges (links). We ran random-walks for $5.6 \cdot 10^8$ restarts, producing a corpus of $4.4 \cdot 10^9$ tokens. We did not optimize any parameter, except the number or restarts, which we run until convergence on RG.

Finally, we used high-dimensional word representations produced with Personalized PageRank (Agirre, de Lacalle, and Soroa 2014), as implemented by UKB, which is publicly available[16]. We used word representations produced from the WordNet and Wikipedia graphs that we just mentioned (PPwn and PPwiki, for short). We ran UKB out-of-the-box, with the default damping value of 0.85. Contrary to RWwn and RWwiki, these word representations have as many dimensions as vertices in the graph, $117 \cdot 10^3$ for PPwn and $3 \cdot 10^6$ for PPwiki.

Combinations

The goal of this experiment is to explore whether simple combination techniques can be used to combine a relatively

is performed using Spearman, which is based on ranks.

[14] http://webscope.sandbox.yahoo.com/

[15] https://code.google.com/p/word2vec/

[16] http://ixa2.si.ehu.eus/ukb

	RG	SL	WSS	WSR	MTU	MEN	WS	sim	rel	all
(a) WBU	76.4	39.7	76.6	61.5	64.6	74.6	67.3	64.2	66.9	64.5
(b) GOOG	76.0	44.2	77.8	60.0	65.5	74.6	68.1	66.0	66.5	65.6
(c) RWwn	82.3	52.5	76.2	58.7	62.1	75.4	68.7	70.3	65.4	68.2
(d) PPVwn	85.7	49.3	69.4	44.1	54.5	66.1	56.9	68.1	54.9	62.5
(e) RWwiki	79.6	32.3	67.5	48.2	43.9	60.9	59.3	59.8	51.0	55.2
(f) PPVwiki	88.6	29.2	80.7	62.1	64.5	74.1	72.7	66.2	66.9	65.8
CAT(ac)	84.2	52.2	83.3	68.0	72.1	80.6	75.3	73.2	**73.6**	72.9
CAT(ace)	**91.2**	51.4	80.4	64.0	66.4	78.4	73.6	74.3	69.6	72.2
CAT(abce)	**91.2**	51.6	80.7	64.2	66.7	78.6	73.8	74.5	69.4	72.4
AVG(ac)	84.4	51.7	82.1	68.0	71.6	80.8	74.7	72.8	73.5	72.7
AVG(ace)	89.5	52.6	82.4	68.2	71.2	81.4	75.9	74.8	73.6	74.1
AVG(abce)	89.0	52.1	83.5	68.2	73.4	81.7	76.5	74.9	74.4	74.5
AVG(-f)	89.4	54.1	84.0	68.6	73.7	82.1	76.9	75.8	74.8	75.2
AVG(-e)	86.4	53.8	83.8	**69.3**	**74.0**	81.8	76.3	74.6	75.0	74.4
AVG(-d)	89.9	52.9	84.0	68.8	73.5	82.0	77.1	75.6	74.7	75.1
AVG(-c)	89.6	51.4	83.9	66.8	70.8	80.6	76.2	75.0	72.7	73.3
AVG(-b)	89.9	55.3	83.7	69.1	71.6	82.0	77.0	76.3	74.3	75.2
AVG(-a)	90.4	**56.6**	83.2	62.7	71.8	81.6	77.1	**76.8**	72.0	75.5
AVG(ALL)	90.2	54.7	**84.3**	69.1	73.7	**82.8**	**77.4**	76.4	**75.1**	**75.7**
s-o-t-a	86.0	55.2	80.0	*70.0*	*75.1*	80.0	*85.0*	73.7	75.0	*76.3*

Table 4: Table showing Spearman performance of single word representations (top rows) and selected combinations using CAT and AVG, including ablation (-x meaning all representations except x) and all methods (ALL). Last row for state-of-the-art. Leftmost columns for single datasets, and rightmost for averages across similarity, relatedness and all datasets. Bold for best results among our combinations, italics for best result for state-of-the-art.

large number of complementary word representations. Due to different dimensionality (e.g. 300 vs. thousands), and the large number of possible combinations, we tried the simple combinations CAT and AVG, but limiting CAT to vectors with the same dimensionality[17].

The top rows in Table 4 show the performance of each word representations in isolation. The best relatedness results are for Wikipedia-based PPV, while the best overall and similarity results are for embeddings of random walks over WordNet.

The following rows in the table show the best results for CAT for each number of systems. While CAT is very effective when combining 2 sources, it fails to improve results with more sources, as 3-way and 4-way combinations perform lower than the best 2-way combination.

We also show some selected combinations for AVG, which include the same representations as shown for CAT, ablation results, and results when all are combined. Although AVG underperforms CAT in 2-way combinations, the more sources it combines the better it performs. In fact, the best results are obtained when averaging over all six methods. The ablation results show that all methods contribute to overall performance, as removing any method from the 6-way combination reduces performance (ablation rows and ALL row in Table 4).

[17]Preliminary results showed that the results with vectors of different dimensionality were very poor.

Finally, the last row shows the best reported results in each of the datasets, as follows: RG (Hassan and Mihalcea 2011), SL (Goikoetxea, Soroa, and Agirre 2015), WSS (Baroni, Dinu, and Kruszewski 2014), WSR (Baroni, Dinu, and Kruszewski 2014), MTU (Halawi et al. 2012), MEN (Bruni, Tran, and Baroni 2014), WS (Halawi et al. 2012). Our combined system improves over the state-of-the-art in RG, SL, WSS and MEN. Note that the sim, rel and all for the last row correspond to the average of each of the best systems (i.e. a non-existing system), and still, our combined system beats them on similarity, equals them in relatedness and attains the same results overall, showing that simple combinations of independently learned representations are a promising avenue of research.

Conclusions

In this paper, we show that a simple concatenation of independently learned embeddings outperforms more complex combination techniques in word similarity and relatedness datasets. The key insight is that a dedicated method based on random walks over WordNet is able to represent WordNet information better than lists of restrictions, as used in retrofitting and other methods. The high performance of simple combinations seems to be enabled by the complementarity of corpus-based and WordNet-based embeddings. In addition, we show that simple averaging of six word representations yields results beyond the state-of-the-art in some of the datasets. All software and data are publicly available, as well as WordNet-based and concatenated embeddings for all words[18].

With respect to methods which try to fold-in information from knowledge-bases into the representation learning method, our results seem to point to an alternative route. Independently learning representation models for rich knowledge-base like WordNet seems to be an interesting research direction, as well as researching on more complex combination methods which are based on independently learned vector spaces.

Acknowledgements

We thank Faruqui for his help running his systems. This work was partially funded by MINECO (CHIST-ERA READERS – PCIN-2013-002-C02-01) and the European Commission (QTLEAP – FP7-ICT-2013.4.1-610516). The IXA group is funded by the Basque Government (A type Research Group). Josu Goikoetxea enjoys a grant from the University of the Basque Country.

References

Agirre, E.; Barrena, A.; and Soroa, A. 2015. Studying the wikipedia hyperlink graph for relatedness and disambiguation. *arXiv preprint arXiv:1503.01655*.

Agirre, E.; Alfonseca, E.; Hall, K.; Kravalova, J.; Paşca, M.; and Soroa, A. 2009. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of of NAACL-HLT*, 19–27.

[18]http://ixa2.si.ehu.es/ukb

Agirre, E.; de Lacalle, O. L.; and Soroa, A. 2014. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics* 40(1):57–84.

Al-Rfou, R.; Perozzi, B.; and Skiena, S. 2013. Polyglot: Distributed word representations for multilingual nlp. *arXiv preprint arXiv:1307.1662*.

Baker, C. F.; Fillmore, C. J.; and Lowe, J. B. 1998. The berkeley framenet project. In *Proceedings of the ACL-Volume 1*, 86–90.

Baroni, M.; Dinu, G.; and Kruszewski, G. 2014. Dont count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of ACL*, volume 1.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD*, 1247–1250.

Bruni, E.; Tran, N.-K.; and Baroni, M. 2014. Multimodal distributional semantics. *JAIR* 49:1–47.

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

Faruqui, M., and Dyer, C. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of EACL*, 462–471.

Faruqui, M.; Dodge, J.; Jauhar, S. K.; Dyer, C.; Hovy, E.; and Smith, N. A. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of NAACL-HLT*, 1606–1615.

Gabrilovich, E., and Markovitch, S. 2007. Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In *IJCAI*, volume 7, 1606–1611.

Ganitkevitch, J.; Van Durme, B.; and Callison-Burch, C. 2013. PPDB: The Paraphrase Database. In *HLT-NAACL*.

Goikoetxea, J.; Soroa, A.; and Agirre, E. 2015. Random Walks and Neural Network Language Models on Knowledge Bases. In *Proceedings of NAACL-HLT*, 1434–1439.

Habash, N., and Dorr, B. 2003. Catvar: A database of categorial variations for english. In *Proc. of the MT Summit*.

Halawi, G.; Dror, G.; Gabrilovich, E.; and Koren, Y. 2012. Large-scale learning of word relatedness with constraints. In *Proceedings of the ACM SIGKDD*, 1406–1414.

Hassan, S., and Mihalcea, R. 2011. Semantic relatedness using salient semantic analysis. In *AAAI*.

Hill, F.; Reichart, R.; and Korhonen, A. 2014. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *arXiv preprint arXiv:1408.3456*.

Huang, E. H.; Socher, R.; Manning, C. D.; and Ng, A. Y. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of ACL*.

Jolliffe, I. 2014. *Principal Component Analysis*. John Wiley & Sons, Ltd.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in NIPS*.

Mikolov, T.; Yih, W.-t.; and Zweig, G. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL-HLT*, 746–751.

Miller, G. A. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.

Minnen, G.; Carroll, J.; and Pearce, D. 2001. Applied morphological processing of english. *Natural Language Engineering* 7(03):207–223.

Napoles, C.; Gormley, M.; and Van Durme, B. 2012. Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, 95–100.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. *Proceedings of EMNLP 2014* 12:1532–1543.

Press, W.; Teukolsky, S.; Vetterling, W.; and Flannery, B. 2002. *Numerical Recipes: The Art of Scientific Computing V 2.10 With Linux Or Single-Screen License*. Cambridge University Press.

Radinsky, K.; Agichtein, E.; Gabrilovich, E.; and Markovitch, S. 2011. A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of WWW*, 337–346.

Rastogi, P.; Van Durme, B.; and Arora, R. 2015. Multiview LSA: Representation Learning via Generalized CCA. In *Proceedings of NAACL-HLT*, 556–566.

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

Scharnhorst, K. 2001. Angles in complex vector spaces. *Acta Applicandae Mathematica* 69(1):95–103.

Socher, R.; Pennington, J.; Huang, E. H.; Ng, A. Y.; and Manning, C. D. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of EMNLP*, 151–161.

Tian, F.; Gao, B.; Chen, E.; and Liu, T.-Y. 2015. Learning better word embedding by asymmetric low-rank projection of knowledge graph. *arXiv preprint arXiv:1505.04891*.

Turian, J.; Ratinov, L.; and Bengio, Y. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of ACL*, 384–394.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014a. Knowledge graph and text jointly embedding. In *Proceedings of EMNLP*, 1591–1601. ACL.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014b. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of AAAI*, 1112–1119. Citeseer.

Wittek, P.; Koopman, B.; Zuccon, G.; and Darányi, S. 2014. Combining word semantics within complex hilbert space for information retrieval. In *Quantum Interaction*. Springer. 160–171.