2019-10-11

Size Matters: The Impact of Training Size in Taxonomically-Enriched Word Embeddings

Alfredo Maldonado
Trinity College Dublin, Ireland, maldonaa@tcd.ie

Filip Klubicka
Technological University Dublin, d17124386@mytudublin.ie

John D. Kelleher
Technological University Dublin, john.d.kelleher@tudublin.ie

Research Article

Alfredo Maldonado*, Filip Klubička, and John Kelleher

Size Matters: The Impact of Training Size in Taxonomically-Enriched Word Embeddings

https://doi.org/10.1515/comp-2019-0009
Received May 17, 2019; accepted Jul 04, 2019

Abstract: Word embeddings trained on natural corpora (e.g., newspaper collections, Wikipedia or the Web) excel in capturing thematic similarity ("topical relatedness") on word pairs such as 'coffee' and 'cup' or 'bus' and 'road'. However, they are less successful on pairs showing taxonomic similarity, like 'cup' and 'mug' (near synonyms) or 'bus' and 'train' (types of public transport). Moreover, purely taxonomy-based embeddings (e.g. those trained on a random-walk of WordNet's structure) outperform natural-corpus embeddings in taxonomic similarity but underperform them in thematic similarity. Previous work suggests that performance gains in both types of similarity can be achieved by enriching natural-corpus embeddings with taxonomic information from taxonomies like WordNet. This taxonomic enrichment can be done by combining natural-corpus embeddings with taxonomic embeddings (e.g. those trained on a random-walk of WordNet's structure). This paper conducts a deep analysis of this assumption and shows that both the size of the natural corpus and of the random-walk coverage of the WordNet structure play a crucial role in the performance of combined (enriched) vectors in both similarity tasks. Specifically, we show that embeddings trained on medium-sized natural corpora benefit the most from taxonomic enrichment whilst embeddings trained on large natural corpora only benefit from this enrichment when evaluated on taxonomic similarity tasks. The implication of this is that care has to be taken in controlling the size of the natural corpus and the size of the random-walk used to train vectors. In addition, we find that, whilst the WordNet structure is finite and it is possible to fully traverse it in a single pass, the repetition of well-connected WordNet concepts in extended random-walks effectively reinforces taxonomic relations in the learned embeddings.

Keywords: word embeddings, taxonomic embeddings, WordNet, semantic similarity, taxonomic enrichment, retrofitting

1 Introduction

Word embeddings are vectors that capture the distributional semantic information of words in the corpora on which they are trained [1, 2]. They have been shown to perform well on thematic similarity[1] benchmarks [3], but have been less successful in stricter taxonomic and synonymic benchmarks [4, 5]. In response, there have been recent efforts to incorporate explicit taxonomic information from lexical taxonomies, such as WordNet [6], into word embeddings [7, 8]. This process usually involves modifying pretrained word embeddings according to constraints placed by the structure of the lexical taxonomy in question. For example, retrofitting [7] introduces an objective function that reduces the distance between vectors that represent words contained in the same WordNet synset.

In addition, there have also been separate efforts to build vectors that directly encode semantic information from lexical taxonomies without referring to textual data. For example, sparse (non-distributional) linguistic vectors [9] have been derived from various knowledge sources (FrameNet, WordNet, etc.) Each dimension in these sparse linguistic vectors represent whether a word belongs to a particular synset, holds a particular taxonomic relation, and so on. Other efforts, by contrast, have sought to construct true distributional embeddings on lexical taxonomies by traversing them in a random-walk fashion [10]. These random-walk taxonomic embeddings outperform natural-corpus embeddings on strict taxonomic similarity benchmarks, such as SimLex-999 [4], a gold standard

***Corresponding Author: Alfredo Maldonado:** ADAPT Centre at Trinity College Dublin, Dublin, Ireland; Email: maldonaa@tcd.ie
Filip Klubička: ADAPT Centre at Technological University Dublin, Dublin, Ireland; Email: filip.klubicka@adaptcentre.ie
John Kelleher: ADAPT Centre at Technological University Dublin, Dublin, Ireland; Email: john.kelleher@adaptcentre.ie

1 Often called semantic or topical "relatedness" in the literature. See Section 3.

focusing on taxonomic/synonymic (rather than thematic) similarity.

It has been proposed that natural-corpus embeddings be combined with taxonomic embeddings as a taxonomic enrichment method [11]. Given that both embedding types can use the same learning algorithm, such as Skip-Gram or CBOW [10], this combination seems to be compatible and natural. In this paper, we study two specific vector combination methods: **concatenation** and **fine-tuning**. **Concatenation** consists of simply concatenating the d-dimensional random-walk taxonomic vector for each word with the d-dimensional natural-corpus vector for that same word into a single vector of dimensionality $2d$ (see Section 5.1). Meanwhile, **fine-tuning** consists in further training natural-corpus embeddings on a pseudo-corpus generated by a random-walk of a taxonomy, essentially injecting taxonomic information in the existing natural-corpus embeddings (see Section 5.2).

In spite of the fact that taxonomic and natural-corpus embeddings can use the same training algorithm, it is important to note that the contexts for target words in both embedding types are categorically different: contexts in natural text are made of naturally co-ocurring words. In contrast, contexts in WordNet random-walks are words that are taxonomically related to the target word (e.g. its hypernym, hyponym, co-hyponym, etc.) We discuss this distinction in more depth in Section 3, but essentially, the kind of contextual information that each set of vectors carry is complementary to each other. As a result, we investigate this complementarity as a means of taxonomic enrichment, comparing it against the original taxonomic enrichment method: retrofitting [7].

The main research question we pose relates to finding the optimal amount of taxonomic and natural-corpus training data needed to obtain performance gains in thematic and synonymic benchmarks. It is well-known that word embeddings in general perform better when large amounts of training data are available to them. However, while it is possible to train on increasingly larger amounts of natural-text data (e.g. by crawling the Web), taxonomies are finite. Nevertheless, it is possible to produce very extensive random-walks across the taxonomy network, thus producing larger amounts of (potentially repetitive) training data. In this paper we conduct experiments combining natural-corpus and taxonomic vectors trained on data of different sizes. For natural-corpus embeddings, we simply use Wikipedia text samples of different sizes. For the taxonomic embeddings, we generate training data of different sizes by conducting random-walks over WordNet of varying durations. We observe that whilst performance on thematic and synonymic benchmarks improves as the training data size increases on both natural-corpus and random-walk embeddings, the latter achieve higher performance in the synonymic benchmark with relatively smaller training data sizes. We also confirm previous studies finding that the performance of concatenated natural-corpus and random-walk embeddings can be superior to their individual performance (when not combined) in thematic benchmarks [11]. Crucially however, we find that this result only holds on embeddings trained on medium-sized natural corpora. An implicit assumption has been that vector combination will always increase performance, i.e. that the higher performing embedding in a concatenation is the floor upon which vector combination will always improve. We demonstrate that this is not always the case; and, based on this finding, give recommendations regarding dataset scenarios when a vector combination is likely to be beneficial, and when it is not.

Lastly, we conduct an analysis of the training data generated by WordNet random-walk. We find that although there is a fair amount of repeated sentences in the larger generated training sets, this repetition does not negatively impact performance; and, in fact, it may reinforce the taxonomic relationships of the concepts learned.

An attractive property of taxonomic random-walk training, is that it can be easily conducted through unmodified, off-the-shelf word embedding training programs (e.g. word2vec). This can be achieved by first generating a pseudo-corpus by crawling the WordNet structure and outputting the lexical items in the nodes visited, and then by running the word embedding training program on the generated pseudo-corpus. Given that good performance can be achieved with relatively small random-walk pseudo-corpora, orders of magnitude smaller than the size of a natural corpus required for comparable performance, the computational requirements of this method are significantly low.

Our code and generated datasets are being made available online.[2]

2 Related work

Previous work focusing on encoding information from knowledge resources through embeddings can be categorised into three broad families: (1) **knowledge-resource encoding** methods that directly learn knowl-

2 https://github.com/GreenParachute/wordnet-randomwalk-python

edge resources, (2) **semantic specialisation** techniques that modify pre-trained vectors in such way so that their cosine similarity ends up measuring a specific semantic relation, and (3) **taxonomic enrichment** approaches that seek to augment the similarity of words in pre-trained corpora, based on their taxonomic relationship as expressed by a knowledge resource (this is in addition to the thematic relations already learned through their original corpus training).

Examples of **knowledge-resource encoding** methods include non-distributional sparse word vectors from lexical resources [9], Poincaré embeddings that represent the structure of the WordNet taxonomy in hyperbolic space [12], and embeddings that encode all semantic relationships expressed in a biomedical ontology within a single vector space [13]. Meanwhile, Agirre et al. [14] follow a stochastic approach based on Personalised Page Rank: they compute the probability of reaching a synset from a target word, following a random-walk on a given WordNet relation. Goikoetxea et al. [10] built upon this work, but instead of computing random-walk probabilities, they used an off-the-shelf implementation of the word2vec Skip-Gram algorithm to train embeddings directly on a random walk of the WordNet taxonomy.

By contrast, examples of the **semantic specialisation** approach are PARAGRAM [15], counter-fitting [16], Hypervec [17], Attract-Repel [18] and the work of Nguyen et al. [19] on synonyms and antonyms. By applying different modifications on the objective function, the aim of these works is to convert the cosine similarity function into a function that measures the specific type of semantic relation learnt, while weighting down the thematic relationship originally learnt during pre-training on a text corpus. More recently, Vulić et al. [20] and Ponti et al. [21] introduced global specialisation models where vectors for words that are missing in the knowledge resource are also updated.

An example of **taxonomic enrichment** is retrofitting pre-trained natural-corpus embeddings by reducing the distance between words that are directly linked in knowledge sources like WordNet [7], MeSH [22] and ConceptNet [23]. In addition, the embeddings produced by the random-walk method introduced by Goikoetxea et al. [10] can be readily combined with natural-corpus embeddings in order to enrich them [11].

The quality of vectors produced by knowledge-resource encoding, semantic specialisation and taxonomic enrichment have been evaluated through diverse semantic similarity benchmarks. These benchmarks include WordSim-353 [24], which conflates taxonomic similarity with thematic similarity, SimLex-999 [4] which focuses on taxonomic similarity and SemEval-17 [25], which considers thematic and taxonomic similarity as two points on a scale of degrees of similarity. See Section 6 for more details on these benchmarks.

Table 1: Spearman scores of a selection of methods on three benchmarks: WordSim-353 (WS), SimLex-999 (SL) and SemEval-2017 (SE). Highest value in each benchmark column is state of the art for that benchmark. Abbreviated methods are:
SG: text embeddings trained via Skip-Gram.
PPR/WN: Personalised Page-Rank over WordNet.
RW/WN: Random-Walk over WordNet.
RW+SG: RW/WN vectors concatenated to SG vectors.
* Evaluated in our experimental reproduction.
** Evaluated by [8] in their experimental reproduction.

Method Type	Method	Ref.	WS	SL	SE
Text	SG	[10]	.69	.44	.57*
Encoding	PPR/WN	[14]	.72	--	--
Encoding	RW/WN	[10]	.70*	.52	.50*
Enrichment	RW+SG	[10]	**.80**	.55	.72*
Enrichment	Retrofitting	[7]	.70	.44*	**.80****
Specialisation	Attract-Repel	[18]	--	**.71**	--

Table 1 shows Spearman correlation scores on WordSim-353, SimLex-999 and SemEval-17 of some state-of-the-art and recent systems that implement the three approach families mentioned earlier. In general, performance tends to be worse on SimLex-999 than on SemEval-17 and WordSim-353. However, notice that Attract-Repel [18] has recently obtained scores as high as 0.71 on SimLex-999. Attract-Repel specialises in learning (and distinguishing from) synonymic and antonymic relations and incorporates information from rich knowledge sources.

Of special note from these results is that Goikoetxea et al. [11] found that simple vector concatenation (RW+SG in Table 1) perform better than retrofitting (and other more complex methods of vector combination) in WordSim-353 and SimLex-999. The original retrofitting method [7], exploited the Paraphrase Database [26], WordNet and FrameNet [27] ontologies. They achieve a Spearman score of 0.70 on the WordSim-353 dataset. However, their work is focused only on using synonyms derived from synsets, and they do not make use of other types of relations found in knowledge bases, such as hypernymy and hyponymy.

The state of the art in SemEval-17 is held by the original winners in this competition, who employed retrofitting in their system [8]. They perform what they call "expanded retrofitting", which means that they use a union of the vocabularies from the corpus embeddings and semantic network, as opposed to regular retrofitting where the vocabularies are intersected. In addition, they use ConceptNet [23] instead of WordNet, and employ heuristics to handle out-of-vocabulary words, such as averaging the vectors of

the neighbours of a given out-of-vocabulary word in the semantic network. With this system, they achieve a Spearman score of 0.80 (Table 1).

Despite the appealing simplicity and strong performance of the embeddings resulting from the concatenation of random-walk and natural corpus embeddings (RW+SG in Table 1), they have received little attention in the literature. One exception is our own work in Klubička et al. [28] where we found that word distributions in random-walk corpora are similar to natural corpora in terms of Zipf's and Heap's law. We also analysed the role of rare words in the performance of the embeddings. However, as that work explores only random-walk pseudo-corpora, the effects of the size of the training data used during training on the random-walk corpus were not explored in depth; only relatively small pseudo-corpus sizes were considered, and no attention is given to natural corpora at all. This is important given that the quality of vectors increases in proportion to the training data size. Also, as mentioned in the introduction, given that the WordNet structure is finite, it remains a question of whether doing very extensive random walks, potentially revisiting the full structure more than once, is beneficial at all. We address these lines of inquiry in this work.

3 Thematic and Taxonomic Contexts

Although semantic relatedness is often treated as a single concept in the literature on lexical semantics, there are at least two different aspects of semantic relatedness: taxonomic and non-taxonomic (e.g. contextual, thematic) relations. This distinction has been described and explored in-depth by Kacmajor and Kelleher [5]. According to their work, **Taxonomic relatedness** is relatedness defined as belonging to the same taxonomic category, which involves having common features and functions. On the other hand, **thematic relatedness** is relatedness existing by virtue of co-occurrence of concepts in any sort of context, and specifically of events or scenarios, which involves performing complementary roles.

This raises the question of what kind of similarity is being modelled, represented and ultimately evaluated in the literature, and whether the correct datasets are used for these tasks. Kacmajor and Kelleher state that related work on semantic relatedness and similarity often does not specify what kind of similarity is being modelled or evaluated, but find that when 'similarity' is used in the literature it most often refers to taxonomic similarity. Yet this distinction is really important to keep in mind, as the ability to differentiate between taxonomic and thematic relations can lead to enhanced statistical language models. They claim that both types of relations are important, but in a different way: thematic relations express high-probability co-occurrences and thus help to predict the next word, while taxonomic relations indicate which words can be replaced by other words.[3] In addition, Kacmajor and Kelleher find that different benchmark evaluation datasets are actually better suited to evaluate one kind of relatedness over the other. They perform experiments to build models that are best at each of the two dimensions of semantic relatedness and those that achieve a good balance between the two.

This distinction between taxonomic and thematic relatedness is an important consideration for us as well, as the aim of this work is to combine the two different axes of semantic relations into one embedding representation.

4 Training Data Generation by WordNet random-walk

In this work we learn taxonomic embeddings over WordNet via random-walk. More specifically, we conduct a random-walk of the WordNet hierarchical structure and produce a pseudo-corpus by emmitting the lexical items of the visited synsets. We then train embeddings using a standard implementation of the Skip-Gram algorithm. This section describes the process to generate the random-walk pseudo-corpus used for training these taxonomic embeddings.

Our pseudo-corpus generation process is inspired by the work of Goikoetxea et al. [10], who performed random walks over WordNet graphs to create synthetic contexts on which word embeddings are trained, thus creating word representations. In this work, the authors treat the WordNet knowledge base as an undirected graph of interlinked synsets and construct an inverse dictionary that maps the synsets to the words (lemmas) that are linked to it. Their method first chooses a synset at random from the set of all synsets, and then performs a random walk starting from it. They use a predefined dampening parameter (α) to determine when to stop the walk. In other words, at each synset, the random walk might move on to a neighbouring synset with probability α, or might terminate with the probability

3 In the linguistics literature, the concepts of taxonomic and thematic relatedness are also referred to as paradigmatic and syntagmatic relations, respectively.

$1 - \alpha$. This dampening factor is usually set to 0.85. Each time the random walk reaches a synset, a lemma belonging to the synset is emitted at random using the probabilities in the inverse dictionary. When the random walk terminates, the sequence of emitted words forms a pseudo-sentence of the pseudo-corpus upon which the word embeddings will be trained, and the process repeats until a predetermined number of sentences or tokens have been generated. The authors do not explicitly state which kinds of semantic relations they traverse during their random walk.

Our re-implementation is largely the same, except that we only traverse hypernymic and hyponymic relationships and ignore other relationship types such as meronym and antonym relations. These are two examples typical of the pseudo-sentences produced by our system:

1. acoustic gramophone Victrola gramophone phonograph machine ATM
2. shatterproof glass glass natural glass

As can be seen, both sentences contain words that hold taxonomic (i.e. hypernymic, hyponymic and co-hyponymic relations) relations among them.

Just as Goikoetxea et al., we treat WordNet's taxonomic relations as an undirected network, and start our walk at a random synset in the taxonomy. Before moving on to the next synset, we choose a lemma corresponding to that synset. Lemmas are chosen based on their probabilities provided by WordNet. The probabilities in the inverse dictionary (the mapping from synsets to lemmas) are available from WordNet itself, but are expressed as frequencies rather than probabilities. We choose one at random based on the probability distribution derived from the frequency counts.

Once the lemma has been emitted, we check if the synset has any hypernym and/or hyponym connections assigned to it. If it does, we choose one at random with equal probability and continue the walk towards it, choosing a new lemma from the new synset. We stop the walk either if (a) there are no more connections to take, or (b) the process is terminated according to the dampening factor α. We then restart the process and create a new pseudo-sentence, until we have generated the required number of sentences.

One important thing to note is that we allow our algorithm to go back to a node that has already been visited, but we do not allow it to choose a lemma that has already appeared in the sentence we are generating at that time. In addition, as opposed to Goikoetxea et al. who produce multiword terms like `Victrola_gramophone`, `shatterproof_glass`, `natural_glass` essentially treating them as words with spaces, we divide them up into their individual constituent words (e.g. `Victrola gramophone`, `shatterproof glass`, `natural glass`). This is why there are repeated words in the example sentences (1 and 2).

5 Methods for Combining Natural-Corpus and Taxonomic Embeddings

In this work we study two natural ways of incorporating information from lexical taxonomies into natural-text word embeddings: 1) concatenation of the natural-text vectors with vectors independently trained on pseudo-corpus generated through random-walk over WordNet, and 2) fine-tuning the natural-text vectors by continuing their training on the random-walk pseudo-corpus. These methods are explained in the following two subsections.

All word embeddings are trained on a corpus (natural or generated by random walk) using a slightly modified version of Pytorch SGNS, a publicly available implementation[4] of the Skip-Gram with Negative Sampling (SGNS) algorithm, introduced by Mikolov et al. [1, 2]. Our modifications mostly concern adding new vectors for words encountered during the fine-tuning step that did not occur in the original natural corpus, as well as other minor data-handling optimisations. The objective function is not modified in any way. Training is conducted for a pre-determined number of epochs. All settings and hyperparameters are described in Section 7.1.

5.1 Concatenation

Concatenation requires word vectors that have been trained on a sufficiently large corpus using a suitable word embedding software package. Separately, it also requires word vectors that have been trained on a pseudo-corpus generated by a random walk of WordNet (or other suitable taxonomy), using the same word embedding software.

The two sets of word vectors are concatenated to form a single set of word vectors. The concatenation process proceeds as follows. A union of the vocabularies from the two sets of word vectors is conducted. If $\mathbf{r_i}$ is a word vector representing word w_i trained on the natural corpus and if $\mathbf{p_i}$ is a word vector also representing w_i but trained on the

4 https://github.com/theeluwin/pytorch-sgns

Table 2: Similarity scale used by human annotators in the SemEval-17 Task 2 challenge. Adapted from [25].

Score	Interpretation	Description	Kind of similarity
4	Very Similar	Synonymous pair (e.g. midday-noon)	Taxonomic
3	Similar	Words in pair share many aspects of meaning with slight differences. They refer to similar but not identical concepts. (e.g. lion-zebra, firefighter-policeman)	Taxonomic
2	Slightly Similar	Words in pair are not very similar but share a topic, domain or function (e.g. house-window, aeroplane-pilot)	Thematic
1	Dissimilar	Words in pair are clearly dissimilar but may share some small details, a far relationship or a domain in common and could be found together in a document on the same topic (software-keyboard, driver-suspension)	Thematic
0	Totally Dissimilar and Unrelated	Words are unrelated and do not share the same topic	Dissimilar

random-walk pseudo-corpus, then the concatenated vector $\mathbf{c_i} = [\mathbf{r_i}; \mathbf{p_i}]$ is constructed. If w_i does not exist in the vocabulary of one of the sets of word vectors, then the centroid of all words for that set is used in its place. For example, $\mathbf{r} = \frac{1}{n}\sum_{j=1}^{n}\mathbf{r_j}$ if there is no representation for w_i in the natural corpus word vector set and $\mathbf{p} = \frac{1}{m}\sum_{j=1}^{m}\mathbf{p_j}$ if the pseudo-corpus does not have a representation for w_i. We interpret this centroid to give a representative flavour of the corpus missing the word (i.e. a fall-back mechanism). An alternative using a vector of zeroes to represent a missing word, performed slightly worse than this centroid in our preliminary experimentation. If $\mathbf{r_i}$ and $\mathbf{p_i}$ are of dimensionality d, the concatenated vectors' dimensionality will be $2d$.

5.2 Fine-tuning

In the fine-tuning workflow, the word vectors trained on the natural corpus are used, but they continue to be trained (fine-tuned) on the random-walk pseudo-corpus. Concretely, the word vectors trained on the natural corpus are loaded as pre-initialised vectors. New, randomly-initialised word vectors are created for any words present in the vocabulary of the random-walk pseudo-corpus but not in the natural-corpus vocabulary. Then, SGNS training is continued on the pseudo-corpus for a pre-determined number of epochs. If the dimensionality of the natural corpus vectors is d, then the dimensionality of the fine-tuned vectors will also be d.

6 Corpora and lexical similarity datasets

In our experiments we train our natural-corpus vectors sentences sampled randomly from the Wikipedia corpus from the Polyglot project[5] [29]. We produce different sets of vectors from samples of the 1, 5, 10, 15 and 20% of sentences from this Wikipedia corpus. The first five rows in Table 3 show the sizes, in terms of sentences, tokens and types, of these Wikipedia samples.

We also generated pseudo-corpora using the WordNet random walk method described in Section 4. The sizes of these pseudo-corpora are presented in the remaining rows of Table 3. As will be discussed in Section 7.1, good performance starts to be observed from around 100k sentences onwards. So these pseudo-corpora need not be massive.

We test our models on three lexical similarity datasets:

– **SemEval-17** [25] consists of a set of 500 pairs of words, multiword expressions (MWEs) and entities in English[6] from a wide range of domains. These 500 pairs are uniformly distributed across a scale of five degrees of similarity that range from total dissimilarity to complete synonymy, with thematic and taxonomic similarities falling at different points along this scale. Importantly, thematic similarity is considered to be at a lower scale than taxonomic similarity. Table 2 summarises the scale used in this challenge. We added the last column to explicitly distinguish the type of similarity indicated at each point in the scale.

5 https://sites.google.com/site/rmyeid/projects/polyglot

6 We concentrate on the monolingual similarity task in English only.

Table 3: Corpus sizes in number of sentences, tokens and types

Corpus Type	Sentences	Tokens	Types
Wiki (1%)	667,575	16,534,730	467,005
Wiki (5%)	3,333,131	82,650,326	1,281,645
Wiki (10%)	6,672,248	165,363,197	1,951,871
Wiki (15%)	10,000,201	247,928,306	2,490,973
Wiki (20%)	13,335,936	330,692,221	2,945,898
WN/RW	1,000	3,541	2,595
WN/RW	1,000	3,591	2,675
WN/RW	10,000	34,691	16,711
WN/RW	30,000	104,736	34,823
WN/RW	50,000	176,020	46,478
WN/RW	70,000	245,730	54,135
WN/RW	100,000	350,435	62,950
WN/RW	150,000	525,174	71,736
WN/RW	200,000	703,827	77,470
WN/RW	300,000	1,052,906	83,516
WN/RW	500,000	1,756,304	88,735
WN/RW	750,000	2,633,072	91,028
WN/RW	1,000,000	3,517,592	92,070
WN/RW	1,500,000	5,274,584	92,826
WN/RW	2,000,000	7,032,270	93,111
WN/RW	2,500,000	8,791,403	93,252
WN/RW	3,000,000	10,546,605	93,327
WN/RW	3,500,000	12,301,532	93,395
WN/RW	4,000,000	14,067,967	93,426
WN/RW	4,500,000	15,824,999	93,446
WN/RW	5,000,000	17,588,303	93,461
WN/RW	658,024,622	83,000,000	93,530
WN/RW	1,308,182,495	165,000,000	93,538
WN/RW	1,966,276,579	248,000,002	93,539
WN/RW	2,624,244,171	331,000,020	93,539

- **WordSim-353**[7] benchmark [24] is an older and more established semantic similarity dataset that conflates thematic and taxonomic similarities. It consists of 353 word pairs.
- **SimLex-999** [4] consists of 999 word pairs whose similarity judgements emphasise taxonomic and synonymic similarity over all other semantic relations, which receive very low similarity scores. Semantic similarity systems tend to perform much worse on SimLex-999 than on mixed thematic-taxonomic benchmarks such as SemEval-17 and WordSim-353 [4].

7 Experiments

7.1 Setup

Word vectors were trained and combined following the methods described in Section 5. The vectors were computed by the SGNS system using a word window of five words to the left and five words to the right of a sliding focus word, without crossing sentence boundaries. Twenty words were randomly selected from the vocabulary based on their frequency as part of the negative sampling step of the training. The frequencies in this weighting were smoothed by raising them to the power of $\frac{3}{4}$ before dividing by the total. All vectors produced by the SGNS system had 300 dimensions. Vectors were trained for 30 epochs. The dampening α parameter for all generated random-walk pseudo-corpora was set to 0.85.

The concatenated vectors were constructed from the vectors trained on the real corpus at the 30th epoch and the set of vectors trained on a WordNet random-walk pseudo-corpus also at the 30th epoch. The fine-tuned vectors are computed by taking the base corpus vectors at the 30th epoch and further training them on one of the random-walk pseudo-corpora for 30 additional epochs. Word vectors are constructed for all types in each random-walk corpora. For the Wikipedia samples, word vectors are computed only for the 100,000 most frequent word types.

The constructed vectors from all models are evaluated by computing cosine scores on the vectors representing each word in every pair from SemEval-17, WordSim-353 and SimLex-999 and computing Spearman's rank correlation coefficient (henceforth Spearman score) between these cosine scores and the gold standard similarity scores from each benchmark. All models train vectors for unigrams only, so if a benchmark word pair contains a MWE, a pseudo-vector for that MWE is constructed by summing the word vectors of its individual words. If a model does not have a vector representation for a word from a word pair, the system does not output a score for that pair, impacting negatively the model's performance.

7.2 Results and discussion

During our preliminary experimentation with the random-walk embeddings, we observed more dramatic jumps in performance at the smaller training size ranges (0-18M tokens) than at the larger side of the scale (18-331M to-

7 http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/

kens). Conversely, we only started noticing significant performance improvements on the natural corpora once we started hitting the half-million tokens mark (approx. 1% of Wikipedia). Because of this, we present our results using two different kinds of plots. One kind focuses on the smaller random-walk training range (0-18 million words), whereas the other kind gives a full picture covering the full size range (0-331 million tokens) for both embedding sets. We describe these two kinds of plots as we present the results.

Figure 1 presents plots of the first kind, concentrating on the smaller random-walk range (0-18 million tokens) but covering almost the full Wikipedia sample range (0-248 million tokens corresponding to 1, 5, 10 and 15% of Wikipedia). Each row of graphs shows the results for a different sample of Wikipedia. The *y* axis in the figure's plots shows Spearman scores on all models against each of the three semantic similarity benchmarks studied: WordSim-353, SimLex-999 and SemEval-17. The *x* axis in these plots represents the size of the generated WordNet random walk pseudo-corpus in millions of tokens. The Spearman score plotted represents the best score (from all training epochs) achieved for a model at that particular WN/RW corpus size. The models (lines in plots) being evaluated are: (1) vectors trained on the natural corpus only (`Wiki`) drawn as thick black lines, (2) vectors trained soley on the `WN/RW` pseudo-corpora only (thick grey lines), (3) the original WordNet `Retrofitting` method by Faruqui et al. [7][8] (dotted thin magenta line), (4) our `Fine-tuning` combination method (solid thin magenta line) and (5) our `Concatenation` combination method (dashed thin magenta line).

Notice that for vectors which do not depend on the size of the `WN/RW` pseudo-corpus (`Wiki` and `Retrofitting`) a constant horizontal line is drawn. Notice as well that pure `WN/RW` results (thick grey line) do not depend on Wikipedia sample size, so the `WN/RW` plots on the top row are identical to their corresponding plots on the subsequent rows.

Figure 1 shows that, in comparison to all other models, `WN/RW` (thick grey line) presents a very strong performance on the three benchmarks. As the random-walk pseudo-corpus size becomes larger, the model becomes stronger.

On the WordSim-353 dataset, `Retrofitting` beats pure `WN/RW`. However, both combination methods (`Fine-tuning` and `Concatenation`) slightly outperforms `Retrofitting` at the larger samples of Wikipedia. This trend is mainly driven by the performance of pure `Wiki` embeddings. At Wikipedia 15%, however, we start seeing that the performance of vector combinations (especially `Fine-tuning`), start taking over the pure `Wiki` model. There is not much difference in performance between `Retrofitting` and our vector combination methods on this dataset.

On the SimLex-999 dataset, the trend is reversed, in the sense that it is the pure `WN/RW` models driving the performance while `Wiki` stays well behind, even as the Wikipedia sample grows from the top plot to the bottom plot. Notice though that in this dataset the `Retrofitting` model tends to fall somewhere in between the pure `WN/RW` models and the pure `Wiki` model. `WN/RW` and our combination methods (`Concatenation` and `Fine-tuning`) are almost indistinguishable from each other on this dataset, but clearly beat `Retrofitting`, especially at the smaller sizes of Wikipedia. Notice as well that this occurs at a relatively small size of the `WN/RW` corpus (less than 5 million tokens), suggesting that the amount of training data needed to obtain relatively good taxonomic information through random-walk embeddings is relatively small. Notice that as the Wikipedia corpus grows larger, our combination methods modestly outperform pure `WN/RW` vectors, suggesting that the extra information provided by a large natural corpus can complement purely taxonomic information.

On the SemEval-17 dataset, the `Concatenation` vectors outperform all others in all Wikipedia sizes (especially on the smaller Wikipedia sample of 1%), with `Retrofitting` following closely behind. This is not surprising: given that taxonomic and thematic similarities are part of the same similarity scale in the SemEval-17 dataset, a mixture of the two types of information will translate in good results.

The performance of all models is lower on SimLex-999 in comparison to WordSim-353 and SemEval-17. However, the scores achieved by the WN/RW model are relatively high for this dataset and it beats all models tested by [4] against this benchmark. The current state of the art score on SimLex-999 is 0.71, achieved by Attract-Repel [18], a system that specialises in learning (and distinguishing from) synonymic and antonymic relations and incorporates information from knowledge sources as diverse as BabelNet, Wikipedia, WordNet, etc. Attract-Repel's authors do not evaluate their system on thematic similarity benchmarks. It is not their focus to do so as they seek to specialise their vectors in synonymic similarity. By contrast, we seek to enrich corpus-based vectors with taxonomic information without affecting their ability to perform well on thematic similarity. We believe the experiments presented here demonstrate that this is feasible. Also, our combina-

8 Using Faruqui et al.'s own implementation: https://github.com/mfaruqui/retrofitting

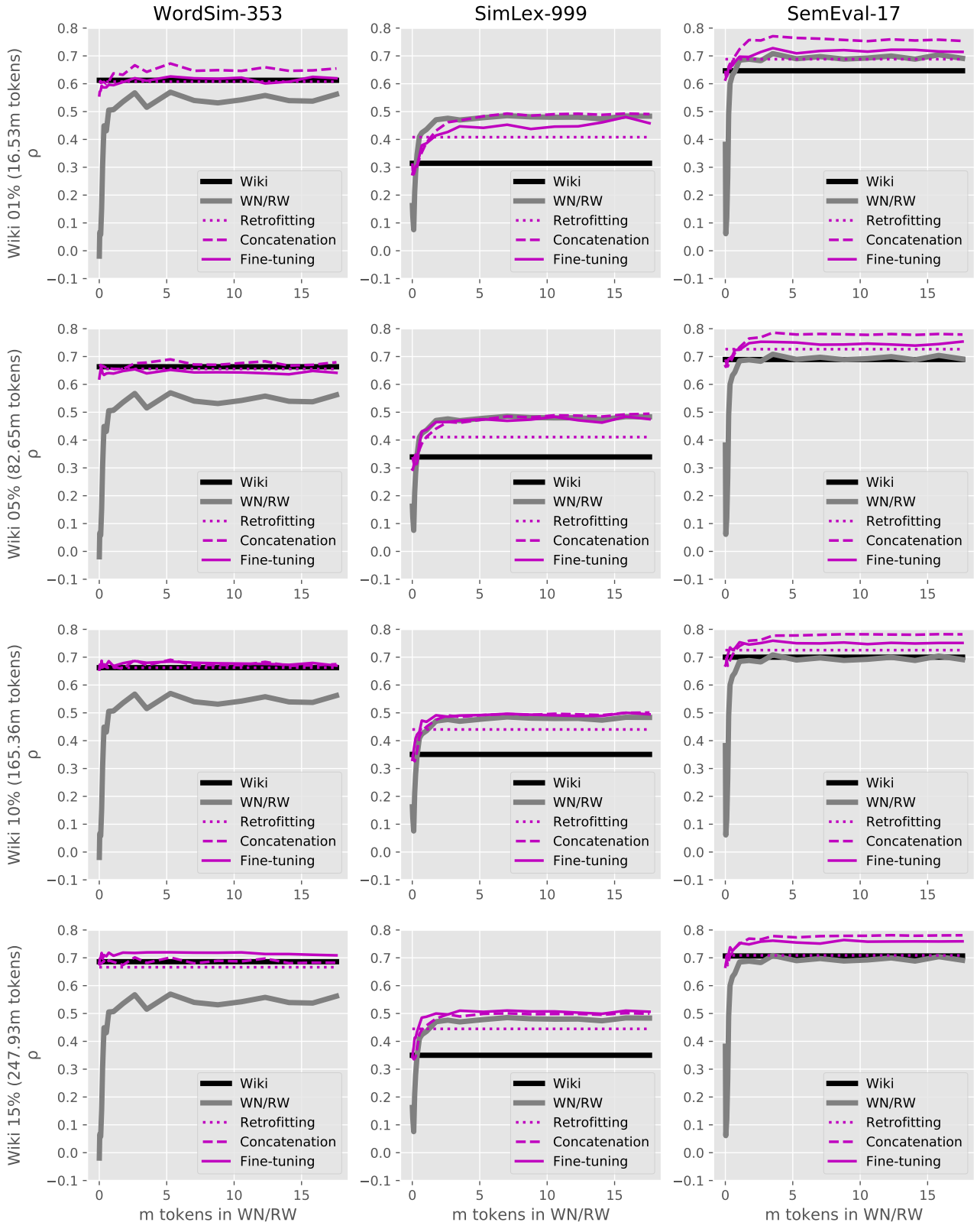


Figure 1: Evaluation results for different training sizes of the Wikipedia (Wiki) and WordNet-via-Random-Walk (WN/RW) corpora, by themselves (solid lines) and in combination via Retrofitting, Concatenation and Fine-tuning. Evaluated using Spearman correlation scores ρ against three manually-annotated datasets: WordSim-353, SimLex-999 and SemEval-17.

Table 4: Vector Type Recommendations based on corpus training sizes for thematic and taxonomic similarity tasks. Random-walk pseudocorpus sizes vary across columns while Natural corpus sizes range across rows. Rough size ranges in tokens: Small: 0-18m, Medium 18-80m, Large: 80m+. We use a star rating system for the top performing vector type, where NR (zero stars) is Not Recommended and should be avoided, * indicates medium performance, ** indicates adequate performance and * * * indicates good performance.

	Thematic Similarity		
RW corpus size / Natural corpus size	**Small**	**Medium**	**Large**
Small	NR	Combination ★	Combination ★
Medium	Natural ★	Combination ★★	Combination ★★
Large	Natural ★★★	Natural ★★★	Natural ★★★

	Taxonomic Similarity		
RW corpus size / Natural corpus size	**Small**	**Medium**	**Large**
Small	NR	RW ★★	RW ★★
Medium	Natural ★	RW ★★	Combination ★★
Large	Natural ★	Combination ★★	Combination ★★

tion methods can easily be scaled to cover additional linguistic resources in general language, in specialised domains and potentially in several languages.

The system that won the official SemEval-17 competition obtained a Spearman score of 0.80 [8, 25]. This system was also a retrofitting system based on the model by [7]. However, instead of WordNet, they employed ConceptNet[9] [23], an ontology containing more complex relationships than WordNet. They also employed some sophisticated heuristics to handle out-of-vocabulary words.

The second kind of plots that we provide are Figure 2 for WordSim-353, Figure 3 for SimLex-999 and Figure 4 for SemEval-17. Each figure contains two plots depicting Spearman scores of our concatenation method: (a) a contour plot of the Spearman scores over the full range of the WordNet RW sizes (x axis) and of the Wikipedia sample sizes (y axis), and (b) a heatmap detailing the numerical Spearman scores over the same range of corpus sizes. Both plots depict the same information. For the contour plots, two zoom-ins are provided: one for low values of WordNet RW sizes (left-hand side), and another for low values of Wikipedia sample sizes (bottom part). This second kind of plots focuses on the concatenation method given that fine-tuning tends to perform similarly as training corpora sizes increase.

For WordSim-353, Figure 2a shows that the best performance is achieved with the largest value of Wikipedia samples and relatively small `WN/RW` sizes (upper-left corner in the main plot). Figure 2b shows this in more detail: on the top row (331m Wikipedia), it can be seen that when `WN/RW` reaches 18 million words, the scores start declining. However, notice that on the rows from 11 million through 17 million, as we move to the left (i.e. as the `WN/RW` corpus increases), the scores significantly improve. This result suggests that combining vectors trained on modestly-sized natural corpora with taxonomic vectors can yield performance increases on thematic similarity. However, vectors trained on large natural corpora do not benefit from this combination with taxonomic vectors. In fact, it could lead to drops in thematic similarity performance.

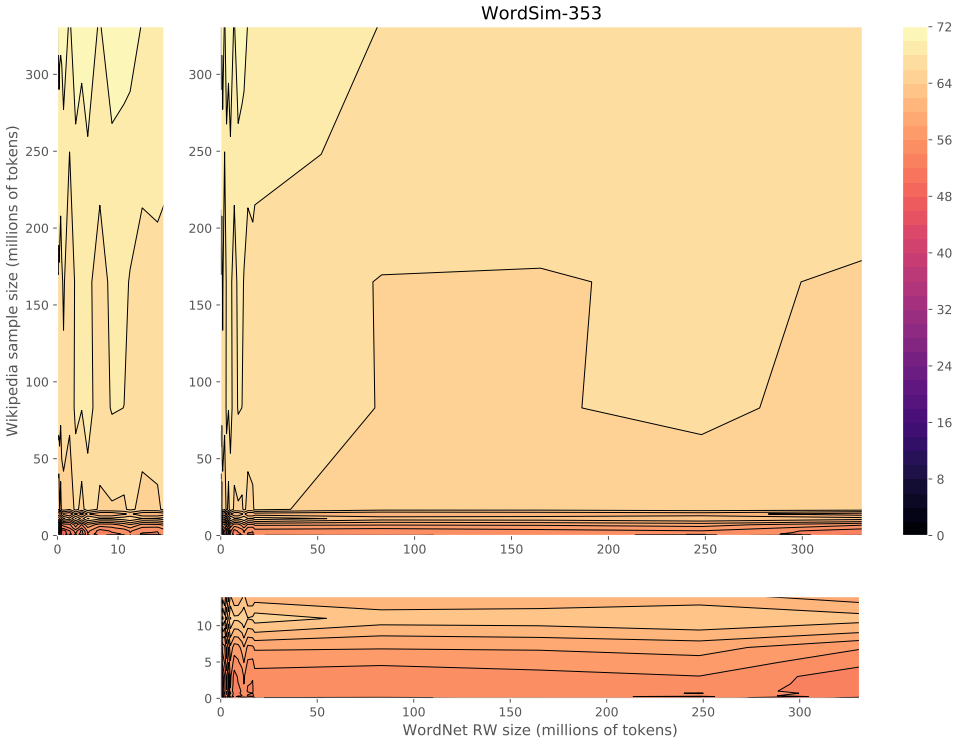
For SimLex-999, Figure 3b shows that the highest performance is achieved with a `WN/RW` model trained on a random-walk of 83 million words and combined with relatively large amounts of a natural corpus (≥17m). Vector combination can give modest improvements over a pure medium-sized (approx. 80m) `WN/RW` model if the natural corpus is very large.

For SemEval-17, Figure 3a shows that good performance can be achieved with both a large natural-corpus and a large random-walk corpus. However, performance starts decreasing towards the very large random-walk embeddings (≥248m). It is not surprising that large amounts of both types of information yield good performance given that the SemEval-17 evaluation scale mixes both types of similarity (Table 2).

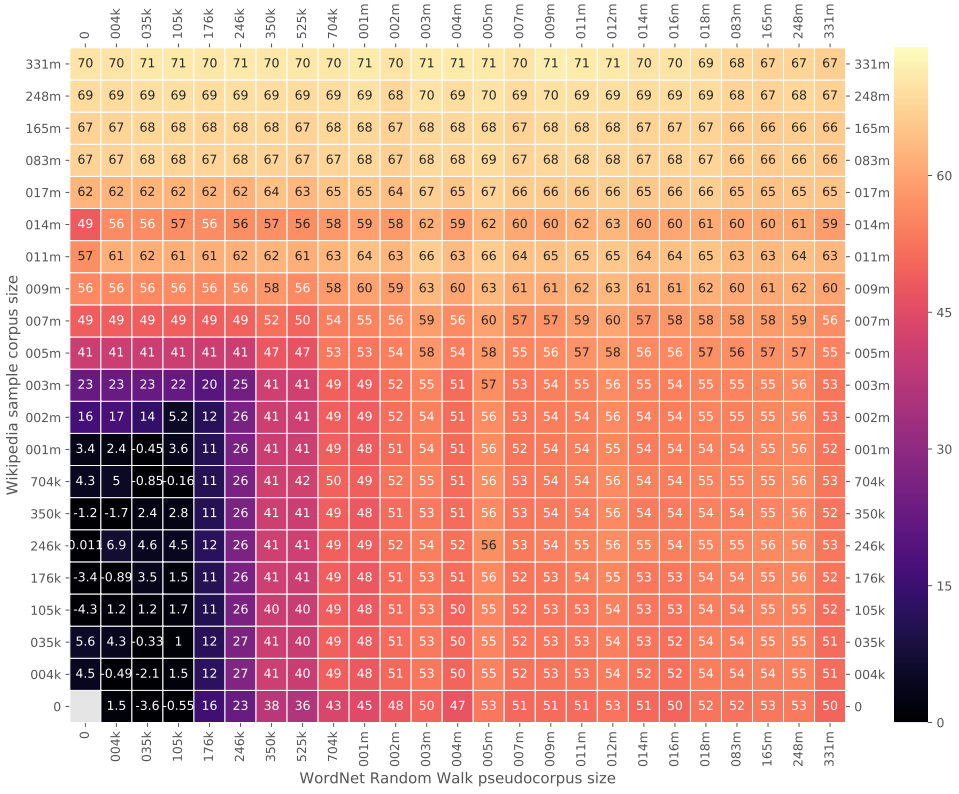
We can summarise the above results into the following two points:

- Thematic similarity is driven mostly by natural-corpus embeddings and not so much by taxonomic embeddings. Enrichment through vector combination, however, can help when natural-corpus vectors are trained on small-to-medium sized corpora. If they are trained on very large corpora, taxonomic enrichment offers no benefit and could actually hinder performance.
- Taxonomic similarity, by contrast, is driven mostly by random-walk vectors. Only medium sizes of random-walk data are needed: there is little benefit to training vectors on very large random walks. Vec-

9 http://conceptnet.io

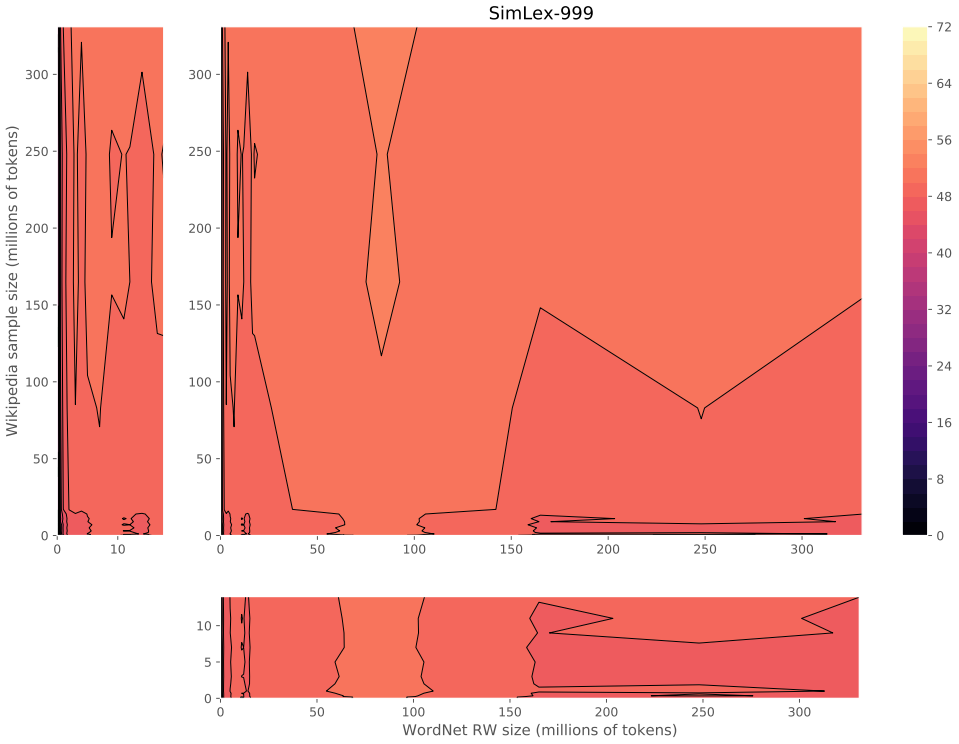


(a) Contour plot of Spearman scores with a zoom-in of lower corpus sizes

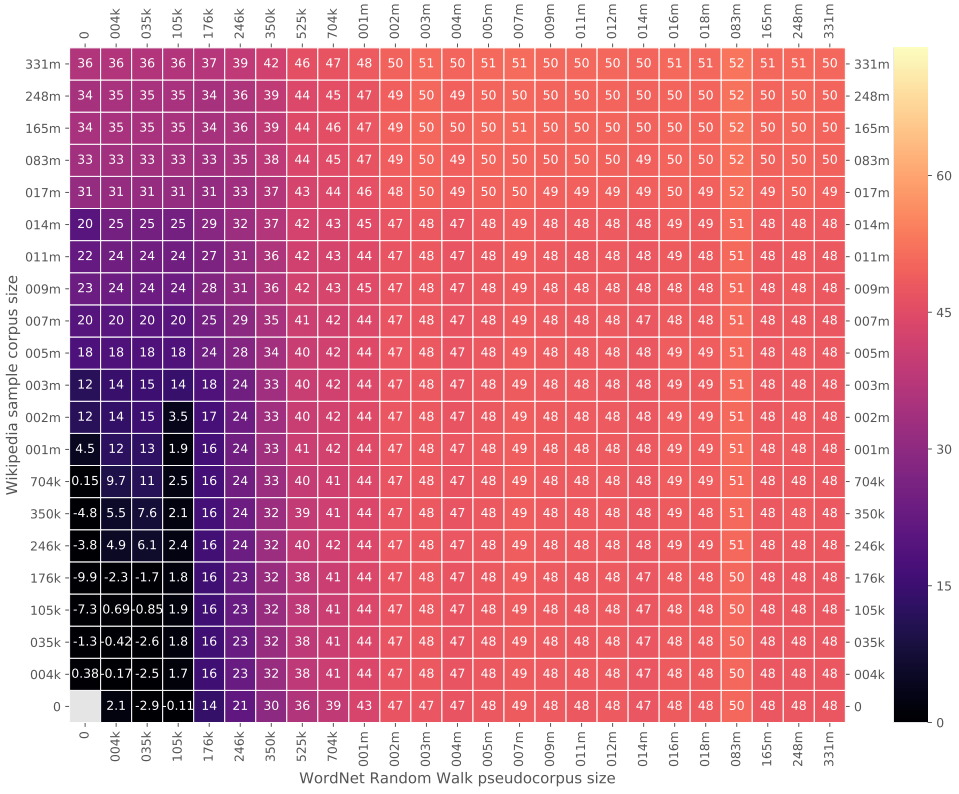


(b) Heatmap with details of same Spearman scores

Figure 2: Spearman scores for the WordSim-353 dataset depicted as a contour plot (2a) and a heatmap (2b)

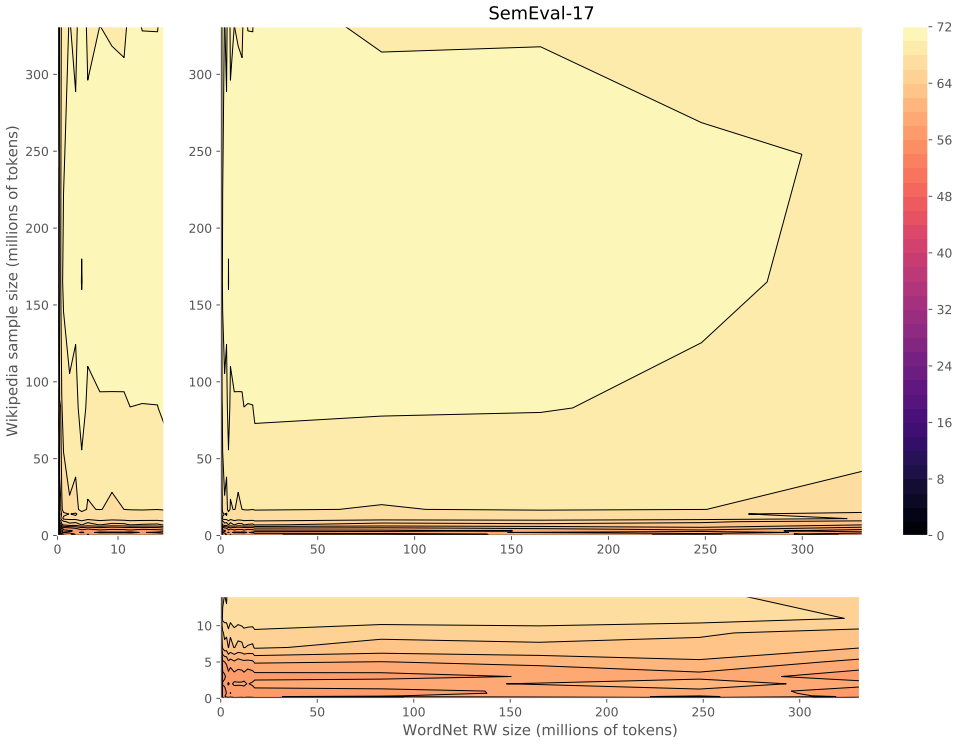


(a) Contour plot of Spearman scores with a zoom-in of lower corpus sizes

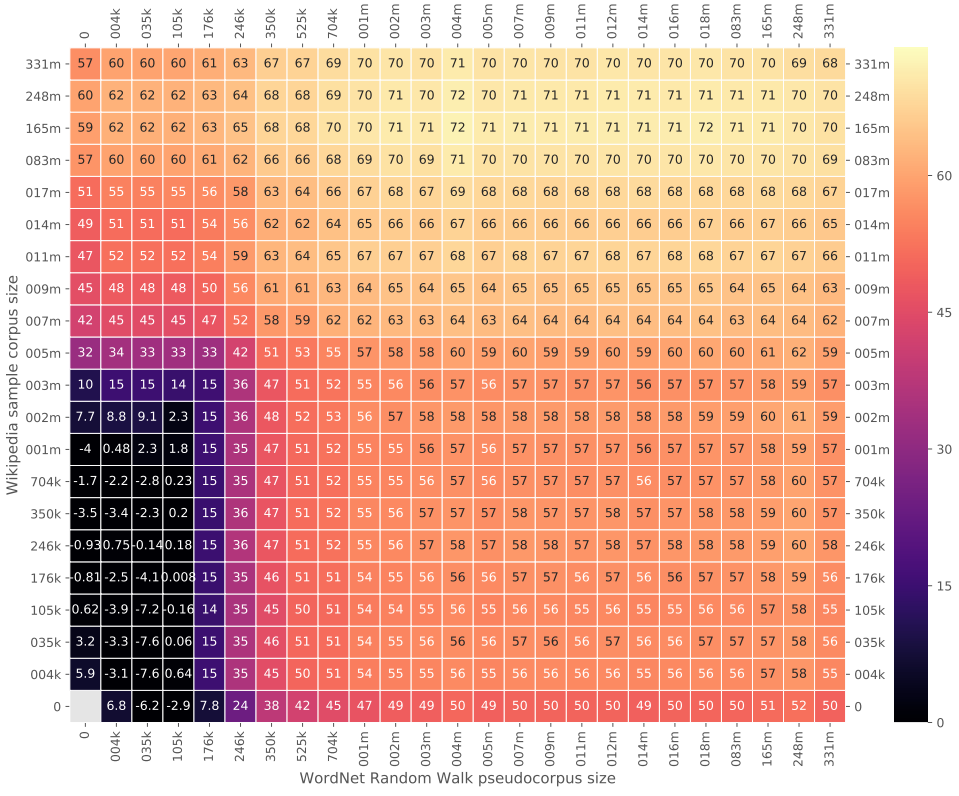


(b) Heatmap with details of same Spearman scores

Figure 3: Spearman scores for the SimLex-999 dataset depicted as a contour plot (3a) and a heatmap (3b)



(a) Contour plot of Spearman scores with a zoom-in of lower corpus sizes



(b) Heatmap with details of same Spearman scores

Figure 4: Spearman scores for the SemEval-17 dataset depicted as a contour plot (4a) and a heatmap (4b)

Table 5: Unique and repeated sentences for each size (in thousands of sentences) of pseudo-corpus

Size k	Unique	Repeated	% Rep.	Size k	Unique	Repeated	% Rep.
1	1000	0	0.00	150	123598	26402	17.60
10	9791	209	2.09	200	158449	41551	20.78
30	28411	1589	5.30	300	221948	78052	26.02
50	46041	3959	7.92	500	335629	164371	32.87
70	62857	7143	10.20	750	461685	288315	38.44
100	86609	13391	13.39	1000	576893	423107	42.31

tor combination is useful when the natural corpus is large.

Table 4 presents a summary of the type of vector we recommend (pure natural-corpus embeddings, pure random-walk embeddings or a combination of the two) depending on the training data size used for each vector type and the type of similarity to be optimised: thematic similarity (top sub-table) or taxonomic similarity (bottom sub-table). The table uses a star rating system to grade a size combination as having either low (no stars), medium (*), adequate (**) or good (* * *) performance. Notice that whilst this star grading is based on the vectors' performance on the similarity tasks described here, the assessment of performance is really application-dependent. Yet, it provides a quick and easy-to-read guide to the properties of vector combinations. Notice also that for taxonomic similarity we consider the best performing vectors as adequate (**). This is due to the fact that newer state-of-the-art vectors tuned on the SimLex-999 dataset obtain much better Spearman scores than those achived by vector concatenation (see Section 2).

7.3 Random-Walk Pseudo-Corpus Analysis

The random-walk pseudo-corpora employed here are relatively modestly-sized and yet are capable of achieving very competitive results. This makes this method computationally affordable. We argue that the strength of the random-walk pseudo-corpora stems from the repetition of words belonging to well connected synsets. There are two ways that a word can be chosen to form part of a pseudo-sentence: 1) by the random synset selection at the start of a pseudo-sentence and 2) by walking from a synset that has a direct (hypernymic or hyponymic) link to the synset containing the word in question. The probability of a word to be selected by (1) is at most 1 over the number of WordNet synsets (i.e. 1/117,659), a considerably rare event. So, the majority of repeated selections of a single word must be done through a walk-in from an adjacent node. If well-connected synsets are being selected repeatedly (mostly through walk-ins), then a significant number of pseudo-sentences will contain more or less the same words. In fact, Table 5 shows the proportion of sentences that contain exactly the same set of words in each of the random-walk pseudo-corpora used in our experiments. Observe that, as expected, as the corpus size increases, the proportion of repeated sentences also increases. Each repetition of a pseudo-sentence effectively reinforces the learning of the taxonomic relations of the words it contains. In other words, because each sentence is generated by a single random walk, and each walk traverses the taxonomy, the set of words that occur within a sentence is dependent on the local topology (connectedness) of the region within the taxonomy the walk traversed. Hence each repeated sentence reinforces the topological relationships within the taxonomy.

8 Conclusion and future work

We analysed two simple methods of vector combination that enrich pre-trained word embeddings with taxonomic information by training on a pseudo-corpus generated by a random walk of the WordNet taxonomy. Vectors trained on random-walk pseudo-corpora are able to encode taxonomic information as demonstrated by their good performance on a synonymic similarity task.

We have demonstrated that taxonomic enrichment of natural-corpus embeddings through vector combination does not always increase the performance of the resulting word embeddings. So care must be taken on how and when this combination should be performed. Table 4 summarises our recommendations on vector combination.

We found that the strength of taxonomic vectors via WordNet random walk comes from the repetition of well-connected WordNet concepts in the generated pseudo-corpora, which effectively reinforces the topological relationships within the taxonomy.

Lastly, relatively good performance can be achieved with modestly-sized random-walk pseudo-corpora, making the usage of our methods computationally affordable.

In this work we focused on one particular type of word embedding model: Skip-Gram. In future work we plan to experiment with other models such as continuous bag-of-words, Glove and more traditional vector-space models of lexical semantics [30]. We also intend to experiment with taxonomies of both general and specialised domains, as well as ontologies that encode other types of semantic relationships.

All the evaluations presented here were intrinsic. We plan to experiment with extrinsic, end-to-end systems of various kinds to evaluate the practical usefulness of enriched vectors in diverse applications.

We would also like to expand our study to several other languages. Finally, we plan to analyse more deeply the statistical and structural properties of the generated random-walk pseudo-corpora from WordNet and make comparisons with random-walk pseudo-corpora from other taxonomies and ontologies.

Acknowledgement: The research in this paper was supported by the ADAPT Centre for Digital Content Technology (https://www.adaptcentre.ie), funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

References

[1] Mikolov T., Corrado G., Chen K., Dean J., Efficient Estimation of Word Representations in Vector Space, in Proceedings of the International Conference on Learning Representations (ICLR 2013), Scottsdale, AZ, 2013, 1–12

[2] Mikolov T., Stutskever I., Chen K., Corrado G., Dean J., Distributed Representations of Words and Phrases and their Compositionality, in Proceedings of the Twenty-Seventh Annual Conference on Neural Information Processing Systems (NIPS) In Advances in Neural Information Processing Systems 26, Lake Tahoe, NV, 2013, 3111–3119

[3] Baroni M., Dinu G., Kruszewski G., 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, MD, 2014, 238–247, 10.3115/v1/P14-1023

[4] Hill F., Reichart R., Korhonen A., SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation, Computational Linguistics, 41(4), 2015, 665–695, 10.1162/COLI

[5] Kacmajor M., Kelleher J. D., Capturing and measuring thematic relatedness, Language Resources and Evaluation, 2019, 1–38, 10.1007/s10579-019-09452-w

[6] Fellbaum C., WordNet: An Electronic Lexical Database, MIT Press, Cambridge, MA, 1998

[7] Faruqui M., Dodge J., Jauhar S. K., Dyer C., Hovy E., Smith N. A., Retrofitting Word Vectors to Semantic Lexicons, in Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, Denver, CO, 2015, 1606–1615, 10.3115/v1/N15-1184

[8] Speer R., Lowry-Duda J., ConceptNet at SemEval-2017 Task 2: Extending Word Embeddings with Multilingual Relational Knowledge, in Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), Vancouver, 2017, 85–89

[9] Faruqui M., Dyer C., Non-distributional Word Vector Representations, in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers), Beijing, 2015, 464–469, 10.3115/v1/P15-2076

[10] Goikoetxea J., Soroa A., Agirre E., Random Walks and Neural Network Language Models on Knowledge Bases, in Human Language Technologies: The 2015 Conference of the North American Chapter of the Association for Computational Linguistics, Denver, CO, 2015, 1434–1439

[11] Goikoetxea J., Agirre E., Soroa A., Single or Multiple? Combining Word Representations Independently Learned from Text and WordNet, in AAAI, 2016

[12] Nickel M., Kiela D., Poincaré Embeddings for Learning Hierarchical Representations, in I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett, eds., Advances in Neural Information Processing Systems 30, Curran Associates, Inc., Long Beach, CA, 2017, 6338–6347

[13] Cohen T., Widdows D., Embedding of semantic predications, Journal of Biomedical Informatics, 68, 2017, 150–166, 10.1016/j.jbi.2017.03.003

[14] Agirre E., Cuadros M., Rigau G., Soroa A., Exploring Knowledge Bases for Similarity., in Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'10), 2010

[15] Wieting J., Bansal M., Gimpel K., Livescu K., Roth D., From Paraphrase Database to Compositional Paraphrase Model and Back, Transactions of the Association for Computational Linguistics, 3, 2015, 345–358

[16] Mrkšić N., Séaghdha D. O., Thomson B., Gašić M., Rojas-Barahona L., Su P. H., Vandyke D., Wen T. H., Young S., Counter-fitting word vectors to linguistic constraints, arXiv preprint arXiv:1603.00892, 2016

[17] Nguyen K. A., Köper M., Schulte im Walde S., Vu N. T., Hierarchical Embeddings for Hypernymy Detection and Directionality, in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, 2017, 233–243

[18] Mrkšić N., Vulić I., Séaghdha D. Ó., Leviant I., Reichart R., Gašić M., Korhonen A., Young S., Semantic Specialisation of Distributional Word Vector Spaces using Monolingual and Cross-Lingual Constraints, Transactions of the Association for Computational Linguistics, 5, 2017, 309–324

[19] Nguyen K. A., Schulte im Walde S., Vu N. T., Integrating Distributional Lexical Contrast into Word Embeddings for Antonym-Synonym Distinction, in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, 2016, 454–459

[20] Vulić I., Glavaš G., Mrkšić N., Korhonen A., Post-Specialisation: Retrofitting Vectors of Words Unseen in Lexical Resources, in Proceedings of NAACL-HLT 2018, New Orleans, LA, 2018, 516–527

[21] Ponti E.M., Vulić I., Glavaš G., Mrkšić N., Korhonen A., Adversarial Propagation and Zero-Shot Cross-Lingual Transfer of Word Vector Specialization, in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, 282–293

[22] Yu Z., Cohen T., Bernstam E. V., Johnson T. R., Wallace B. C., Retrofitting Word Vectors of MeSH Terms to Improve Semantic Similarity Measures, in Proceedings of the Seventh International Workshop on Health Text Mining and Information Analysis (LOUHI), Austin, TX, 2016, 43–51

[23] Speer R., Havasi C., Representing General Relational Knowledge in ConceptNet 5, in Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), Istanbul, 2012, 3679—-3686

[24] Finkelstein L., Gabrilovich E., Matias Y., Rivlin E., Solan Z., Wolfman G., Ruppin E., Placing search in context: the concept revisited, ACM Transactions on Information Systems, 20(1), 2002, 116–131, 10.1145/503104.503110

[25] Camacho-Collados J., Pilehvar M. T., Collier N., Navigli R., SemEval-2017 Task 2: Multilingual and Cross-lingual Semantic Word Similarity, in Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Vancouver, 2017, 15–26

[26] Ganitkevitch J., Van Durme B., Callison-Burch C., PPDB: The paraphrase database, in Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2013, 758–764

[27] Baker C. F., Fillmore C. J., Lowe J. B., The berkeley framenet project, in Proceedings of the 17th international conference on Computational linguistics-Volume 1, Association for Computational Linguistics, 1998, 86–90

[28] Klubička F., Maldonado A., Kelleher J., Synthetic, yet natural: Properties of WordNet random walk corpora and the impact of rare words on embedding performance, in Proceedings of GWC2019: 10th Global WordNet Conference, 2019

[29] Al-Rfou R., Perozzi B., Skiena S., Polyglot: Distributed Word Representations for Multilingual NLP, in Proceedings of the Seventeenth Conference on Computational Natural Language Learning, Sofia, 2013, 183–192, 10.1007/s10479-011-0841-3

[30] Turney P. D., Pantel P., From Frequency to Meaning: Vector Space Models of Semantics, Journal of Artificial Intelligence Research, 37, 2010, 141–188