Top-down word embedding: Using etymology in learning

Pablo Estrada, Kyomin Jung
pablo@snu.ac.kr, kjung@snu.ac.kr
Seoul National University, Korea

Abstract

We propose a novel approach to learn word embeddings based on an extended version of the distributional hypothesis. Our model derives word embedding vectors using the etymological composition of words, rather than the context in which they appear. Our model has the strength of not requiring a large text corpus, but instead it requires reliable access to etymological roots of words, making it specially fit for languages with logographic writing systems.

The model consists on three steps: (1) building an etymological graph, which is a bipartite network of words and etymological roots, (2) obtaining the biadjacency matrix of the etymological graph and reduce its dimensionality with SVD, (3) using columns/rows of the resulting matrices as embedding vectors for their corresponding words.

We test our model in two languages: a set of 117,000 Chinese words, and a set of 135,000 Sino-Korean words. In both cases we show that our model performs well in the task of synonym discovery.

1 Introduction

One important area of research in the field of natural language processing (NLP) is that of word embedding. This consists on using a text corpus to characterize and embed words into rich high-dimensional vector spaces. By mining a text corpus, it is possible to embed words in a continuous space where semantically similar words are embedded close together, and different dimensions can express different semantic characteristics. This is important because by encoding words into vectors, it is possible to represent semantic properties of these words in a way that is more expressive and useful for natural language processing. Word embedding has become an active area of research, and it has been effectively used for sentiment analysis (Chen et al. (2013)), machine translation (Zou et al. (2013)), and other tasks.

The basic idea behind all methods of word embedding is the distributional hypothesis (Levy and Goldberg (2014)). This is the idea that words that appear in similar contexts must express similar ideas: "A word is characterized by the company it keeps". Based on this idea, count-based methods such as LSA (Dumais (2004)), and predictive methods that use neural networks to learn the embedding vectors (Mikolov et al. (2013); Baroni et al. (2014)) were developed, and used in research with success.

In this work, we propose a new approach to learn word embeddings that is based on the etymological roots of words, rather than the context in which they appear in text or speech. Our approach relies on the fact that a shared etymological root between two words expresses a deliberate semantic similarity between these two words; and by leveraging information on these semantic similarities, it is possible to derive the embedding vectors of words. This is akin to extending the distributional hypothesis to consider etymological context as well as textual context: words that appear in similar *etymological* contexts must also express similar concepts.

Based on this hypothesis, we propose an approach that consists on building a graph that captures these etymological relationships, and reducing the dimensionality of its adjacency matrix to learn word

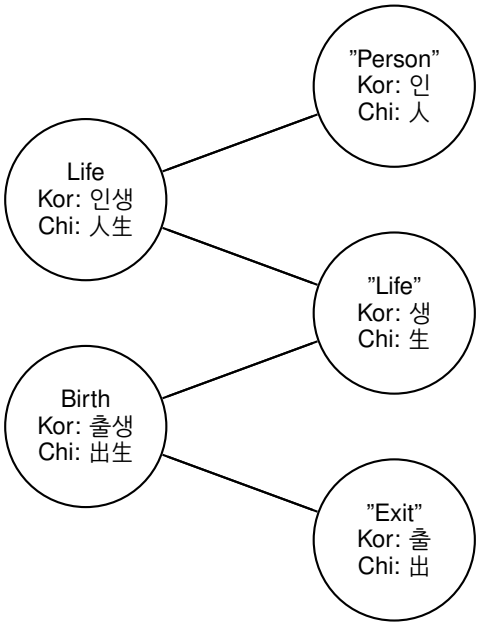


Figure 1: Subset of bipartite graph. On the left are the 'words' that are formed by mixing 'roots', which can be seen on the right. In Sino-Korean vocabulary, most words are formed by using two etymological roots, but there are also words formed by three roots, and a few that are formed by more than three. In Chinese vocabulary, words tend to be longer on average, and thus have a larger amount of etymological roots.

embeddings. To verify the validity of the word embeddings learned by our model we use the task of synonym discovery, whereby we analyze if it's possible to identify a pair of words as synonyms only through their embedding vectors. Synonym discovery is an important task in fields such as information retrieval, and sentiment analysis; and it has been used before to test word embedding schemes (Chen et al. (2013)).

Since our work relies on etymology, it requires a reliable way to obtain the etymological roots of words. In languages with primarily phonetic writing systems, inferring the etymological roots of words is a significant challenge that requires intellectual work to trace words back to their ancestors. This is perhaps the reason that not much research has been made in the data mining community that is based on etymology. This stands in contrast to languages with logographic writing systems, where a word carries morphological information in its writing. This makes the task of etymology extraction much simpler in these languages. This is why our approach is particularly well suited for the Chinese language, and the subset of the Korean vocabulary that is comprised by Sino-Korean words (i.e. Korean words that have been borrowed from Chinese).

Written Chinese is comprised by a large set of *Hanzi*, or characters. Generally, one character represents one syllable of spoken Chinese; and it may represent a word, or be part of a polysyllabic word. The characters themselves can be composed to form new, more complex, characters. Chinese writing has also been adopted in other languages such as Korean, Japanese and formerly also Vietnamese. In this work, we use each character as an *etymological root* that forms part of a word (which is either mono- or polysyllabic); and we study Chinese vocabulary in Korean and in the Chinese language.

1.1 Previous work

There exists limited research on etymological networks in the English language. Particularly (Hunter and Singh (2015)), and (Hunter et al. (2014)) use an etymological network-based approach to study movie scripts and movie reviews in English. They find that an etymological network built through either the script, or the reviews of a film can be used to extract important keywords about the film.

When it comes to work that studies the Chinese language (or the Chinese writing system when used

in other languages) that is based on network science, a popular topic in previous work is to study how radicals combine to form more complex characters (Li and Zhou (2007)). Some studies have created networks based on word co-occurrence (Zhou et al. (2008)). We found only one study that creates a network based on how characters mix to form words (Yamamoto and Yamazaki (2009)). All these studies have studied the graph-theoretical properties of their networks, but have not attempted knowledge extraction from the graphs.

The task of synonym discovery in Chinese vocabulary, which we use to test the appropriateness of our word embeddings, has been tackled in previous work (Lu Yong (2008); Yong et al. (2010)). These studies use a large corpus from the Chinese Wikipedia, and identify synonyms by building a graph where words are linked if their Wikipedia articles link to each other. These studies do not report their performance in general, instead reporting some identified synonym pairs.

In our own previous work, we defined an etymological graph-based framework, and used it in a supervised classification scheme to find pairs of Chinese characters (e.g. etymological roots) that were synonyms (E. and Jung (2016)). In this paper we showed that the etymological graph approach can be effectively used to extract knowledge from a complex etymological network.

Word embedding was defined originally in (Bengio et al. (2006)), where the authors use a neural network-based approach to generate a language model of which word embeddings are a byproduct. Since then, numerous studies have been written where both neural networks and count-based models have been used to produce word embeddings (Mikolov et al. (2013); Baroni et al. (2014)). Aligned embeddings have also been used for machine translation, particularly (Zou et al. (2013)) attempts translation between English and the Chinese language.

To the best of our knowledge, there are no papers that explore any data mining task based on etymology in either languages with phonetic alphabets or with logographic alphabets.

1.2 Our work and contributions

In this work we define a framework to build etymological networks in any language, and use it to build two different networks: One of Sino-Korean words, and one of Chinese words. We then use these graphs to extract semantic knowledge from their underlying network structure.

We extract semantic knowledge by obtaining word embedding vectors from the biadjacency matrix of these etymological graphs. For this we use Singular Value Decomposition (SVD). We also explore the impact of the number of dimensions may have in the amount of useful information that can be conveyed by the embedding vectors, and on the time it takes our model to run and obtain the embedding vectors.

In both the languages that we attempt our experimental task, embedding vectors of synonyms are shown to have significant relationships - unlike those of randomly selected pairs of words. Therefore we show that an etymology-based approach can be used to learn word embeddings, and that the distributional hypothesis can be extended to consider also etymological contexts.

We also compare the performance of our word embedding vectors in he task of synonym discovery against another set of embedding vectors that was constructed with a co-occurrence model.

The code for this paper, as well as the datasets and instructions on how to replicate this work are openly available [1].

2 Method

2.1 Building the etymological graph

An etymological graph is a bipartite network with two sets of nodes: one that represents the roots of the words in a language, while the other set represents the words themselves. In an etymological graph, two nodes are connected if one node represents an etymological root of the word represented by the other, as shown in Figure1.

To build an etymological graph, one may start from a list of words annotated with their etymological roots. By iterating over the list, and iterating over the roots of each word; it is possible to add nodes

[1] https://github.com/pabloem/hanja-graph

and edges to the graph in order. By following this procedure, one will end up with a fully constructed bipartite etymological graph. This procedure is expressed in algorithm 1.

Algorithm 1 Building etymological graph

Require: Empty graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
Require: List of words $\mathcal{W}$ annotated with etymological roots.
for each $w \in \mathcal{W}$ **do**
  $\mathcal{V} \leftarrow \mathcal{V} \bigcup \{w\}$
  for each $root \in w$ **do**
    if $root \notin \mathcal{V}$ **then**
      $\mathcal{V} \leftarrow \mathcal{V} \bigcup \{root\}$
    end if
    $\mathcal{E} \leftarrow \mathcal{E} \bigcup \{\{root, w\}\}$
  end for
end for

As part of our research, we built two graphs using data collected by crawling an online dictionary for the set of Sino-Korean vocabulary; and the ADSO dataset for Chinese vocabulary. Some statistics about these graphs are shown on Table 1. It is interesting to note that the distributions over word length in Chinese is different than in Korean. This is, perhaps, due to the differences in the ways Chinese loan-words are used in the Korean language, and the ways Chinese uses its own words. These differences should not affect the outcome of our model, because they do not affect the construction of the graph.

Table 1: Statistics from the Sino-Korean vocabulary graph, and the Chinese vocabulary graph.

Language	**Chinese**	**Korean**
Num. of Words	117,568	136,045
Num. of Characters	5,115	5,972
Avg. word length	3.36	2.56
Avg. degree of a root-node	76.45	58.2
Words by length		
1 character	2,082	-
2 characters	25,001	77,891
3 characters	35,108	40,024
4 characters	39,249	18,130
5 characters	16,128	-

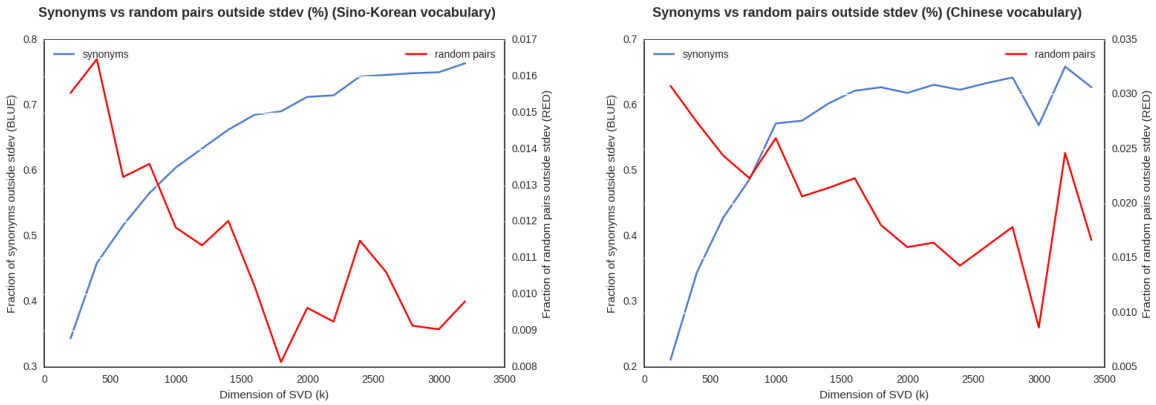
2.2 Learning word embeddings

To obtain the word embeddings from the graphs, truncated Singular Value Decomposition(SVD) was applied to their biadjacency matrices (Asratian et al. (1998)). We use SVD inspired by the techniques of LSA (Dumais (2004)), where it's possible to map words and documents to 'hidden concepts'.

The biadjacency matrix A of a bipartite graph is a matrix of size $n \times m$ where each column represents a node from one bipatite set, and each row represents a node from the other bipartite set. In the case of etymological graphs, each row represents a root node, while each column represents a word node; therefore the matrix A has dimension $\#roots \times \#words$.

By applying SVD, we attempt to approximate the biadjacency matrix A as the product of three matrices $U\Sigma V^*$, where Σ is a diagonal matrix with the k largest singular values in the diagonal, and the matrices U and V^* are matrices of size $\#roots \times k$ and $k \times \#words$ respectively; where k is the dimension into which we chose to reduce matrix A. We use the dimension-reduced column vectors in V^* as embeddings for each word in our vocabulary.

Another matrix decomposition technique worth considering for future work is CUR factorization



(a) Sino-Korean vocabulary

(b) Chinese vocabulary

Figure 2: Proportion of random pairs of words where the dot product of their embedding vectors is far from zero (BLUE) and proportions of pairs of synonyms where the dot product of their embedding vectors is far from zero (RED). Only 1% of the random pairs place far from zero, while about 73%(a) and 65%(b) of synonyms are far from zero.

(Boutsidis and Woodruff (2014)). We're specially interested in its sparsity-maintaining characteristic; since large matrices such as ours can be managed more easily if they are sparse - and SVD eliminates the sparsity of our source matrices.

2.3 Verifying the word embeddings: Synonym discovery

To verify the validity of the embeddings, we selected the task of synonym discovery. To assess whether two words are synonyms, we measure their similarity as proposed in (Blondel et al. (2004)). We expect synonyms to show similarity score above a threshold, which we decide by studying the distribution of the similarity between random pairs of words. In other words, we obtain the dot product of vectors from random pairs of words, and compare them to the dot product of vectors from pairs of synonyms. As random pairs of words are expected to have little semantic relationship, the dot product of their embedded vectors is expected to be close to 0; while the dot product of vectors representing pairs of synonyms is expected to be far from 0 due to the semantic similarity between pair of synonyms, which should be expressed by their embedding in a vector space. For comparison, we used the dataset of Chinese word embeddings that was released as part of (Zou et al. (2013)), which contains embeddings of Chinese words in 50 dimensions. We used this data set on the same task: Synonym discovery by measuring their similarity score as the internal product between vectors.

To obtain the 'ground truth' of synonym pairs, we collected pairs of synonyms from online dictionaries for both Chinese and Sino-Korean vocabulary. In Korean we collected a total of 38,593 pairs of synonyms, while in Chinese we collected 45,731 pairs.

Another task that is often used when working with word embeddings is that of pluralization. We judged this task to be inappropriate for our model, since pluralization in Chinese-based languages is expressed as part of the sentece, and not as part of a word itself.

3 Results

3.1 Performance of synonym discovery task

Experiments show that we were able to reliably identify pairs of synonyms by comparing the dot product between embeddings of pairs of synonyms in both the languages that we tested. Performance was specially good in the Korean language graph, as can be seen in Figure3, where we plot distributions of dot product between random pairs and pairs of synonyms. As shown in the figure, up to 70% of all synonyms have a similarity measure that places them outside the range covered by 99% of random pairs of synonyms. Figure2 helps drive this point by showing the variation of the proportion of synonyms that

are placed outside the standard deviation of the distribution of dot products of embeddings of random pairs of words when we vary the dimension of our embeddings. Interestingly enough, only about 1% of random pairs of words appear outside of this range, and the vast majority of them consistently concentrated around zero. We found that increasing the number of dimensions does reduce this proportion, albeit slightly.

Our embeddings also proved to perform better than our benchmark dataset. Figure3(f) shows the distribution of the similarity measure between pairs of synonyms and random pairs of characters in the benchmark dataset. In this sample, almost 32% of synonyms show a similarity score that places them away from zero, while 5% of random pairs of words are placed outside of that range. Table 2 compares the performance, and the dimensionality of both strategies to learn embeddings.

Table 2: Performance for each model and language

Model	**Language**	**Dimensions**	**Correctly classified synonyms**	**Misclassified random pairs**
Our model	Korean	2000	70%	1%
Our model	Chinese	2000	64%	1.5%
Zou, *et al.*	Chinese	50	32%	5%

3.2 Computation speed of our model

An interesting feature of word embedding models based on matrix factorizations is that training time can be significantly shorter when compared with the time it may take to train a multi-layered neural network. For dimensions under 500, SVD can run very quickly, but as the dimension rises, the factorization step becomes significantly slower. Our model reaches its best performance at around 2000 dimensions, for which the matrix factorization takes over 5 minutes of computation.

Code for our model was developed in Python 3. Paticularly, we used the NetworkX python package to manage and analyze our graphs (Schult and Swart (2008)), and the SciPy (Jones et al. (2001)) and NumPy (Van Der Walt et al. (2011)) libraries to work with matrices and vectors. Our code ran on a Intel Core i7-4790 clocked at 3.60GHz and 16 MB of RAM. Table 3 shows the running time of the factorization of both our graphs and different values for the dimension of the matrix decomposition.

Table 3: Running time of matrix factorization

	Time (seconds)	
SVD k	Sino-Korean vocabulary	Chinese vocabulary
200	6.3	4.6
600	36.7	29.2
1000	71.3	56.9
1400	144.9	113.36
1800	266.2	215.7
2200	407.2	312.4
2400	484.6	337.2
2600	566.4	346.2
3000	737.4	369.1

These running times stand in contrast with the rather large times it takes to train a neural network model. Nonetheless, given that our embeddings require a higher number of dimensions to be effective, SVD on the dimension that we require has a relatively slow performance of up to 10 minutes.

4 Discussion

In this work, we have presented a model to learn word embeddings based on etymology. We have shown that such model can capture semantic information derived from a complex etymological network. Its performance was remarkably good in the task of synonym discovery, even surpassing the performance of an earlier dataset derived from a co-occurrence model. We believe it can also perform well in other tasks.

A noticeable difference between our word embeddings and existing ones is that ours require a much higher number of dimensions to perform well in synonym discovery. Publicly available datasets with word embeddings provide vectors with 25,50 and 100 dimensions (Chen et al. (2013)); but our embeddings reach their highest effectiveness at around 2,000 dimensions. This is likely a consequence of our data being very sparse: while words in word co-occurrence models can have an almost limitless set of contexts in which they appear, words in etymological graphs have a defined number of etymological roots. All the words in our graphs are formed by 5 characters or less.

The approach covered in this paper also has some particular quirks that stem from the use of *historical* (i.e. etymological) data. This is because the meaning of words is not static, but rather evolves with time and use. Word embeddings that are learned from co-occurrence models are able to capture the ways in which words are used in contemporary language. This is not captured by a top-down model that is based on etymology. Our approach would capture the semantics of words as they were *intended* to be used, rather than how they are used in everyday speech.

Our model also does not rely on very large text corpi, though instead it requires reliable access to etymological roots of words. Etymological dictionaries already capture some of this data, but languages continue to evolve and words to be coined at an ever faster pace, so perhaps techniques of machine learning will have to be used to obtain reliable access to etymological roots.

We believe that our model can help expand our understanding of word embedding; and also help us reevaluate the value of etymology in data mining and machine learning. We are excited to see etymological graphs used in other ways to extract knowledge, and hope to see research addressing this. We also are especially interested in seeing this model applied to different languages, such as those with phonetic writing systems.

References

Armen S Asratian, Tristan MJ Denley, and Roland Häggkvist. *Bipartite graphs and their applications*, volume 131. Cambridge University Press, 1998.

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

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

Vincent D Blondel, Anahí Gajardo, Maureen Heymans, Pierre Senellart, and Paul Van Dooren. A measure of similarity between graph vertices: Applications to synonym extraction and web searching. *SIAM review*, 46(4):647–666, 2004.

Christos Boutsidis and David P Woodruff. Optimal cur matrix decompositions. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 353–362. ACM, 2014.

Yanqing Chen, Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. The expressive power of word embeddings. *arXiv preprint arXiv:1301.3226*, 2013.

Susan T Dumais. Latent semantic analysis. *Annual review of information science and technology*, 38(1): 188–230, 2004.

Pablo E. and Kyomin Jung. Knowledge extraction through etymological networks: Synonym discovery in sino-korean words. In *Proceedings of the 5th International conference on Information and Knowledge Management (ICIKM 2016)*, 2016.

Starling Hunter et al. A novel method of network text analysis. *Open Journal of Modern Linguistics*, 4 (02):350, 2014.

Starling David Hunter and Saba Singh. A network text analysis of fight club. *Theory and Practice in Language Studies*, 5(4):737, 2015.

Eric Jones, Travis Oliphant, and Pearu Peterson. Scipy: Open source scientific tools for python. *http://www. scipy. org/*, 2001.

Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, pages 2177–2185, 2014.

Jianyu Li and Jie Zhou. Chinese character structure analysis based on complex networks. *Physica A: Statistical Mechanics and its Applications*, 380:629–638, 2007.

Hou Hanqing Lu Yong. Research on automatic acquiring of chinese synonyms from wiki repository. 3: 287–290, Dec 2008.

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

Daniel A Schult and P Swart. Exploring network structure, dynamics, and function using networkx. In *Proceedings of the 7th Python in Science Conferences (SciPy 2008)*, volume 2008, pages 11–16, 2008.

Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

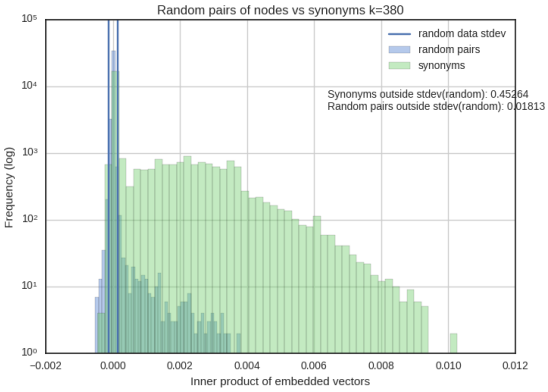
Ken Yamamoto and Yoshihiro Yamazaki. A network of two-chinese-character compound words in the japanese language. *Physica A: Statistical Mechanics and its Applications*, 388(12):2555–2560, 2009.

Lu Yong, Zhang Chengzhi, and Hou Hanqing. Using multiple hybrid strategies to extract chinese synonyms from encyclopedia resources [j]. *Journal of Library Science in China*, 1:010, 2010.

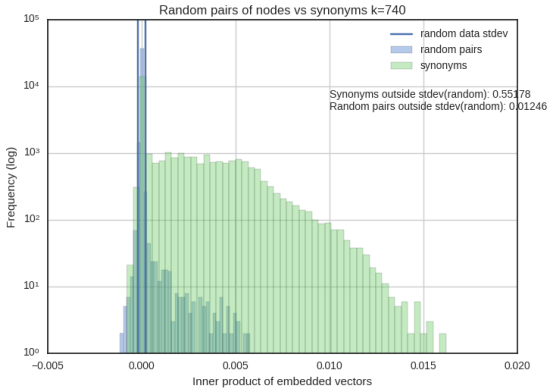
Shuigeng Zhou, Guobiao Hu, Zhongzhi Zhang, and Jihong Guan. An empirical study of chinese language networks. *Physica A: Statistical Mechanics and its Applications*, 387(12):3039–3047, 2008.

Will Y Zou, Richard Socher, Daniel M Cer, and Christopher D Manning. Bilingual word embeddings for phrase-based machine translation. In *EMNLP*, pages 1393–1398, 2013.

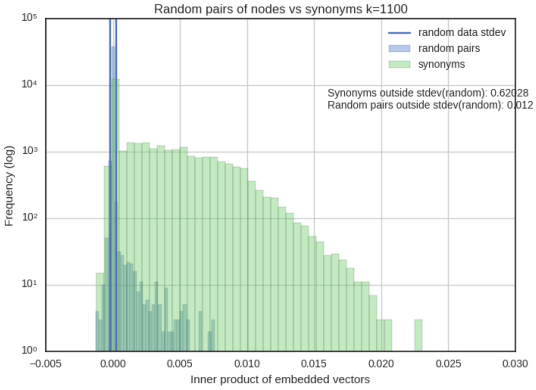
5 Appendix: Distribution of dot-product of synonym pairs and random pairs



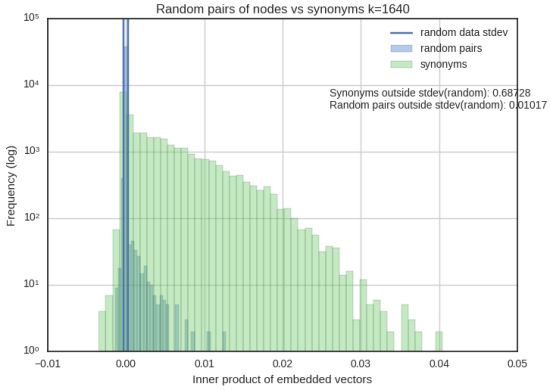
(a) k=380



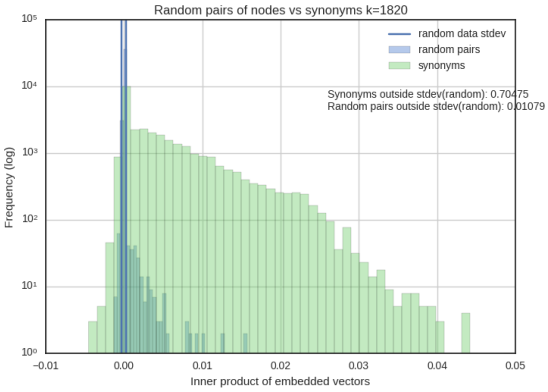
(b) k=740



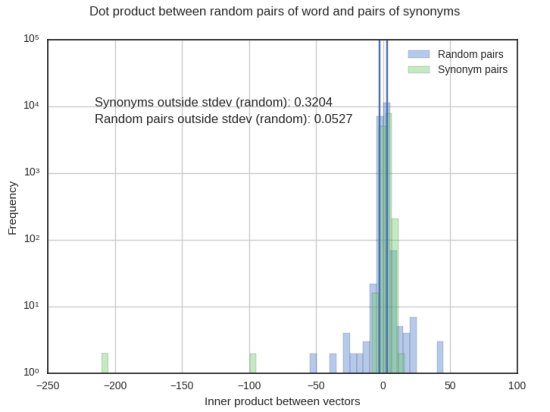
(c) k=1100



(d) k=1640



(e) k=1820



(f) Result from embeddings released from Zou et al. (2013)

Figure 3: Log-scale histograms of dot product between embedding vectors between (green) pairs of synonyms, and (blue) random pairs of words. The vertical lines (blue) represent the standard deviation of the distribution of dot products between random pairs of words.