# Learning Lexical Embeddings with Syntactic and Lexicographic Knowledge

**Tong Wang**
University of Toronto
tong@cs.toronto.edu

**Abdel-rahman Mohamed**
Microsoft Research
asamir@microsoft.com

**Graeme Hirst**
University of Toronto
gh@cs.toronto.edu

## Abstract

We propose two improvements on lexical association used in embedding learning: factorizing individual dependency relations and using lexicographic knowledge from monolingual dictionaries. Both proposals provide low-entropy lexical co-occurrence information, and are empirically shown to improve embedding learning by performing notably better than several popular embedding models in similarity tasks.

## 1 Lexical Embeddings and Relatedness

Lexical embeddings are essentially real-valued distributed representations of words. As a vector-space model, an embedding model approximates semantic relatedness with the Euclidean distance between embeddings, the result of which helps better estimate the real lexical distribution in various NLP tasks. In recent years, researchers have developed efficient and effective algorithms for learning embeddings (Mikolov et al., 2013a; Pennington et al., 2014) and extended model applications from language modelling to various areas in NLP including lexical semantics (Mikolov et al., 2013b) and parsing (Bansal et al., 2014).

To approximate semantic relatedness with geometric distance, objective functions are usually chosen to correlate positively with the Euclidean similarity between the embeddings of related words. Maximizing such an objective function is then equivalent to adjusting the embeddings so that those of the related words will be geometrically closer.

The definition of relatedness among words can have a profound influence on the quality of the resulting embedding models. In most existing studies, relatedness is defined by co-occurrence within a window frame sliding over texts. Although supported by the *distributional hypothesis* (Harris, 1954), this definition suffers from two major limitations. Firstly, the window frame size is usually rather small (for efficiency and sparsity considerations), which increases the false negative rate by missing long-distance dependencies. Secondly, a window frame can (and often does) span across different constituents in a sentence, resulting in an increased false positive rate by associating unrelated words. The problem is worsened as the size of the window increases since each false-positive $n$-gram will appear in two subsuming false-positive $(n+1)$-grams.

Several existing studies have addressed these limitations of window-based contexts. Nonetheless, we hypothesize that lexical embedding learning can further benefit from (1) factorizing syntactic relations into individual relations for structured syntactic information and (2) defining relatedness using lexicographic knowledge. We will show that implementation of these ideas brings notable improvement in lexical similarity tasks.

## 2 Related Work

Lexical embeddings have traditionally been used in language modelling as distributed representations of words (Bengio et al., 2003; Mnih and Hinton, 2009) and have only recently been used in other NLP tasks. Turian et al. (2010), for example, used embeddings from existing language models (Collobert and Weston, 2008; Mnih and Hinton, 2007) as unsupervised lexical features to improve named entity recognition and chunking. Embedding models gained further popularity thanks to the simplicity and effectiveness of the `word2vec` model (Mikolov et al., 2013a), which implicitly factorizes the *point-wise mutual information* matrix shifted by biases consisting of marginal counts of individual words (Levy and Goldberg, 2014b). Efficiency is greatly improved by approximating the computationally costly softmax function with negative sampling (similar to that of Collobert and Weston 2008) or hierarchical softmax (similar to that of Mnih and Hinton 2007).

To address the limitation of contextual locality in many language models (including `word2vec`), Huang et al. (2012) added a "global context score" to the local *n*-gram score (Collobert and Weston, 2008). The concatenation of word vectors and a "document vector" (centroid of the composing word vectors weighted by *idf*) was used as model input. Pennington et al. (2014) proposed to explicitly factorize the global co-occurrence matrix between words, and the resulting log bilinear model achieved state-of-the-art performance in lexical similarity, analogy, and named entity recognition.

Several later studies addressed the limitations of window-based co-occurrence by extending the `word2vec` model to predict words that are *syntactically* related to target words. Levy and Goldberg (2014a) used syntactically related words *non-discriminatively* as syntactic context. Bansal et al. (2014) used a training corpus consisting of sequences of labels following certain manually compiled patterns. Zhao et al. (2014) employed coarse-grained classifications of contexts according to the hierarchical structures in a parse tree.

Semantic relations have also been explored as a form of lexical association. Faruqui et al. (2015) proposed to retrofit pre-trained embeddings (derived using window-based contexts) to semantic lexicons. The goal is to derive a set of embeddings to capture relatedness suggested by semantic lexicons while maintaining their resemblance to the corresponding window-based embeddings. Bollegala et al. (2014) trained an embedding model with lexical, part-of-speech, and dependency patterns extracted from sentences containing frequently co-occurring word pairs. Each relation was represented by a pattern representation matrix, which was combined and updated together with the word representation matrix (i.e., lexical embeddings) in a bilinear objective function.

# 3 The Proposed Models

## 3.1 Factorizing Dependency Relations

One strong limitation of the existing dependency-based models is that no distinctions are made among the many different types of dependency relations. This is essentially a compromise to avoid issues in model complexity and data sparsity, and it precludes the possibility of studying individual or interactive effects of individual dependency relations on embedding learning.

Consequently, we propose a *relation-dependent model* to predict dependents given a governor under *individual* dependency relations. For example, given a nominal governor *apple* of the *adjective modifier* relation (`amod`), an embedding model will be trained to assign higher probability to observed adjectival dependents (e.g., *red*, *sweet*, etc.) than to rarely or never observed ones (e.g., *purple*, *savoury*, etc.). If a model is able to accurately make such predictions, it can then be said to "understand" the meaning of *apple* by possessing semantic knowledge about its certain attributes. By extension, similar models can be trained to learn the meaning of the governors in other dependency relations (e.g., adjectival governors in the inverse relation `amod`$^{-1}$, etc.).

The basic model uses an objective function similar to that of Mikolov et al. (2013a):

$$\log \sigma(\mathbf{e}_g^T \mathbf{e}'_d) + \sum_{i=1}^{k} \mathbf{E}_{\hat{d}_i}[\log \sigma(-\mathbf{e}_g^T \mathbf{e}'_{\hat{d}_i})],$$

where $\mathbf{e}_*$ and $\mathbf{e}'_*$ are the target and the output embeddings for the corresponding words, respectively, and $\sigma$ is the sigmoid function. The subscripts $g$ and $d$ indicate whether an embedding correspond to the governor or the dependent of a dependency pair, and $\hat{d}_*$ correspond to random samples from the dependent vocabulary (drawn by unigram frequency).

## 3.2 Incorporating Lexicographic Knowledge

Semantic information used in existing studies (Section 2) either relies on specialized lexical resources with limited availability or is obtained from complex procedures that are difficult to replicate. To address these issues, we propose to use monolingual dictionaries as a simple yet effective source of semantic knowledge. The defining relation has been demonstrated to have good performance in various semantic tasks (Chodorow et al., 1985; Alshawi, 1987). The inverse of the defining relation (also known as the *Olney Concordance Index*, Reichert et al. 1969) has also been proven useful in building lexicographic taxonomies (Amsler, 1980) and identifying synonyms (Wang and Hirst, 2011). Therefore, we use both the defining relation and its inverse as sources of semantic association in the proposed embedding models.

Lexicographic knowledge is represented by adopting the same terminology used in syntactic dependencies: definienda as governors and definientia as dependents. For example, *apple* is related to *fruit* and *rosaceous* as a governor under `def`, or to *cider* and *pippin* as a dependent under `def`$^{-1}$.

### 3.3 Combining Individual Knowledge Sources

Sparsity is a prominent issue in the relation-dependent models since each individual relation only receives a limited share of the overall co-occurrence information. We also propose a post-hoc, *relation-independent* model that combines the individual knowledge sources. The input of the model is the structured knowledge from relation-dependent models, for example, that *something* can be *red* or *sweet*, or it can *ripen* or *fall*, etc. The training objective is to predict the *original word* given the relation-dependent embeddings, with the intuition that if a model is trained to be able to "solve the riddle" and predict that this *something* is an *apple*, then the model is said to possess generic, relation-independent knowledge about the target word by learning from the relation-dependent knowledge sources.

Given input word $w_I$, its relation-independent embedding is derived by applying a linear model $M$ on the concatenation of its relation-dependent embeddings ($\tilde{\mathbf{e}}_{w_I}$). The objective function of a relation-independent model is then defined as

$$\log\sigma(\mathbf{e}'^T_{w_I} M\tilde{\mathbf{e}}_{w_I}) + \sum_{i=1}^{k} \mathbf{E}_{\bar{w}_i}[\log\sigma(-\mathbf{e}'^T_{\bar{w}_i} M\tilde{\mathbf{e}}_{w_I})],$$

where $\mathbf{e}'_*$ are the context embeddings for the corresponding words. Since $\tilde{\mathbf{e}}_{w_I}$ is a real-valued vector (instead of a 1-hot vector as in relation-dependent models), $M$ can no longer be updated one column at a time. Instead, updates are defined as:

$$\frac{\partial}{\partial M} = [1-\sigma(\mathbf{e}'^T_{w_O} M\tilde{\mathbf{e}}_{w_I})]\mathbf{e}'_{w_O}\tilde{\mathbf{e}}^T_{w_I} - \sum_{i=1}^{k}[1-\sigma(-\mathbf{e}'^T_{w_i} M\tilde{\mathbf{e}}_{w_I})]\mathbf{e}'_{w_i}\tilde{\mathbf{e}}^T_{w_I}.$$

Training is quite efficient in practice due to the low dimensionality of $M$; convergence is achieved after very few epochs.[1]

Note that this model is different from the non-factorized models that conflate multiple dependency relations because the proposed model is a deeper structure with pre-training on the factorized results (via the relation-dependent models) in the first layer.

## 4 Evaluations

### 4.1 Training Data and Baselines

The *Annotated English Gigaword* (Napoles et al., 2012) is used as the main training corpus. It contains 4 billion words from news articles, parsed by the Stanford Parser. A random subset with 17 million words is also used to study the effect of training data size (dubbed *17M*).

Semantic relations are derived from the definition text in the *Online Plain Text English Dictionary*[2]. There are approximately 806,000 definition pairs, 33,000 distinct definienda and 24,000 distinct defining words. The entire corpus has 1.25 million words in a 7.1MB file.

Three baseline systems are used for comparison, including one non-factorized dependency-based model `DEP` (Levy and Goldberg, 2014a) and two window-based embedding models `w2v` (or `word2vec`, Mikolov et al. 2013a) and `GloVe` (Pennington et al., 2014). Embedding dimension is 50 for all models (baselines as well as the proposed). Embeddings in the window-based models are obtained by running the published software for each of these systems on the Gigaword corpus with default values for all hyper-parameters except for vector size (50) and minimum word frequency (100 for the entire Gigaword corpus; 5 for the *17M* subset). For the `w2v` model, for example, we used the skip-gram model with the default value 5 as window size, negative sample size, and epoch size, and 0.025 as initial learning rate.

### 4.2 Lexical Similarity

**Relation-Dependent Models**

Table 1 shows the results on four similarity datasets: *MC* (Miller and Charles, 1991), *RG* (Rubenstein and Goodenough, 1965), *FG* (or *wordsim353*, Finkelstein et al. 2001), and *SL* (or *SimLex*, Hill et al. 2014b). The first three datasets consist of nouns, while the last one also includes verbs (*SL*$_v$) and adjectives (*SL*$_a$) in addition to nouns (*SL*$_n$). Semantically, *FG* contains many related pairs (e.g., *movie–popcorn*), whereas the other three datasets are purely similarity oriented.

[1] We also experimented with updating the relation-dependent embeddings together with $M$, but this worsened evaluation performance.

[2] `http://www.mso.anu.edu.au/~ralph/OPTED/`

| Model | *MC* | *RG* | *FG* | $SL_n$ | $SL_v$ | $SL_a$ |
|---|---|---|---|---|---|---|
| `amod` | **.766** | **.798** | .572 | **.566** | .154 | .466 |
| `amod`$^{-1}$ | .272 | .296 | .220 | .218 | .248 | **.602** |
| `nsubj` | .442 | .350 | .376 | .388 | **.392** | .464 |
| `nn` | .596 | .620 | .514 | .486 | .130 | .068 |
| Baselines | | | | | | |
| `DEP` | .640 | .670 | .510 | .400 | .240 | .350 |
| `w2v` | .656 | .618 | **.600** | .382 | .237 | .560 |
| `GloVe` | .609 | .629 | .546 | .346 | .142 | .517 |

Table 1: Correlation between human judgement and *cosine* similarity of embeddings (trained on the Gigaword corpus) on six similarity datasets.

Performance is measured by *Spearman's* $\rho$ between system scores and human judgements of similarity between the pairs that accompany each dataset.

When dependency information is factorized into individual relations, models using the best-performing relation for each dataset[3] out-perform the baselines by large margins on 5 out of the 6 datasets. In comparison, the advantage of the syntactic information is not at all obvious when they are used in a non-factorized fashion in the `DEP` model; it out-performs the window-based methods (below the dashed line) on only 3 datasets with limited margins. However, the window-based methods consistently outperform the dependency-based methods on the *FG* dataset, confirming our intuition that window-based methods are better at capturing relatedness than similarity.

When dependency relations are factorized into individual types, sparsity is a rather prominent issue especially when the training corpus is small. With sufficient training data, however, factorized models consistently outperform all baselines by very large margins on all but the *FG* dataset. Average correlation (weighted by the size of each sub-dataset corresponding to the three POS's) on the *SL* dataset is 0.531, outperforming the best reported result on the dataset (Hill et al., 2014a).

| Model | *MC* | *RG* | *FG* | $SL_n$ | $SL_v$ | $SL_a$ |
|---|---|---|---|---|---|---|
| Rel. Dep. #1 | .512 | .486 | .380 | .354 | .222 | .394 |
| Rel. Dep. #2 | .390 | .380 | .360 | .304 | .206 | .236 |
| Rel. Indep. | **.570** | .550 | .392 | **.360** | **.238** | .338 |
| Baselines | | | | | | |
| `DEP` | .530 | **.558** | .506 | .346 | .138 | .412 |
| `w2v` | .563 | .491 | **.562** | .287 | .065 | .379 |
| `GloVe` | .306 | .368 | .308 | .132 | −.007 | .254 |

Table 2: Lexical similarity performance of relation-independent models (trained on the *17M* corpus) combining top two best-performing relations for each POS.

Although the co-occurrence data is sparse, it is nonetheless highly "focused" (Levy and Goldberg, 2014a) with much lower entropy. As a result, convergence is much faster when compared to the non-factorized models such as `DEP`, which takes up to 10 times more iterations to converge.

Among the individual dependency relations, the most effective relations for nouns, adjectives, and verbs are `amod`, `amod`$^{-1}$, and `nsubj`, respectively. For nouns, we observed a notable gap in performance between `amod` and `nn`. Data inspection reveals that a much higher proportion of `nn` modifiers are proper nouns (64.0% compared to about 0.01% in `amod`). The comparison suggests that, as noun modifiers, `amod` describes the attributes of nominal concepts while `nn` are more often instantiations, which apparently is semantically less informative. On the other hand, `nn` is the better choice if the goal is to train embeddings for proper nouns.

### Relation-Independent Model

The relation-independent model (Section 3.3) is implemented by combining the top two best-performing relations for each POS: `amod` and `dobj`$^{-1}$ for noun pairs, `nsubj` and `dobj` for verb pairs, and `amod`$^{-1}$ and `dobj`$^{-1}$ for adjective pairs.

Lexical similarity results on the *17M* corpus are listed in Table 2. The combined results improve over the best relation-dependent models for all categories except for $SL_a$ (adjectives), where only the top-performing relation-dependent model (`amod`$^{-1}$) yielded statistically significant results and thus, results are worsened by combining the second-best relation-dependent source `dobj`$^{-1}$ (which is essentially noise). Comparing to baselines, the relation-independent model achieves better results in four out of the six cat-

[3] We did not hold out validation data to choose the best-performing relations for each dataset. Our assumption is that the dominant part-of-speech of the words in each dataset is the determining factor of the top-performing syntactic relation for that dataset. Consequently, the choice of this relation should be relatively constant without having to rely on traditional parameter tuning. For the four noun datasets, for example, we observed that `amod` is consistently the top-performing relation, and we subsequently assumed similar consistency on the verb and the adjective datasets. The same observations and rationales apply for the relation-independent experiments.

| Model | *MC* | *RG* | *FG* | $SL_n$ | $SL_v$ | $SL_a$ |
|---|---|---|---|---|---|---|
| def | .640 | .626 | .378 | .332 | .320 | .306 |
| def$^{-1}$ | .740 | .626 | .436 | .366 | .332 | .376 |
| Combined | **.754** | **.722** | .530 | **.410** | **.356** | .412 |
| w2v | .656 | .618 | **.600** | .382 | .237 | **.560** |

Table 3: Lexical similarity performance of models using dictionary definitions and compared to word2vec trained on the Gigaword corpus.

egories.

### Using Dictionary Definitions

Embeddings trained on dictionary definitions are also evaluated on the similarity datasets, and the results are shown in Table 3. The individual relations (defining and inverse) perform surprisingly well on the datasets when compared to word2vec. The relation-independent model brings consistent improvement by combining the relations, and the results compare favourably to word2vec trained on the entire Gigaword corpus. Similar to dependency relations, lexicographic information is also better at capturing similarity than relatedness, as suggested by the results.

## 5 Conclusions

This study explored the notion of relatedness in embedding models by incorporating syntactic and lexicographic knowledge. Compared to existing syntax-based embedding models, the proposed embedding models benefits from factorizing syntactic information by individual dependency relations. Empirically, syntactic information from individual dependency types brings about notable improvement in model performance at a much higher rate of convergence. Lexicographic knowledge from monolingual dictionaries also helps improve lexical embedding learning. Embeddings trained on a compact, knowledge-intensive resource rival state-of-the-art models trained on free texts thousands of times larger in size.

## Acknowledgments

We thank Gerald Penn, Ruslan Salakhutdinov, Suzanne Stevenson, and Xiaodan Zhu for their insightful comments, as well as the anonymous reviewers for their valuable feedback. This study is financially supported by the Natural Sciences and Engineering Research Council of Canada.

## References

Hiyan Alshawi. Processing dictionary definitions with phrasal pattern hierarchies. *Computational Linguistics*, 13(3-4):195–202, 1987.

Robert Amsler. *The structure of the Merriam-Webster Pocket Dictionary*. PhD thesis, The University of Texas at Austin, 1980.

Mohit Bansal, Kevin Gimpel, and Karen Livescu. Tailoring continuous word representations for dependency parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 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, 2003.

Danushka Bollegala, Takanori Maehara, Yuichi Yoshida, and Ken-ichi Kawarabayashi. Learning word representations from relational graphs. *arXiv preprint arXiv:1412.2378*, 2014.

Martin Chodorow, Roy Byrd, and George Heidorn. Extracting semantic hierarchies from a large on-line dictionary. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, pages 299–304, Chicago, Illinois, USA, 1985.

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

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, Colorado, 2015. Association for Computational Linguistics.

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

Zellig Harris. Distributional structure. *Word*, 10 (23):146–162, 1954.

Felix Hill, Kyunghyun Cho, Sebastien Jean, Coline Devin, and Yoshua Bengio. Embedding word similarity with neural machine translation. *arXiv preprint arXiv:1412.6448*, 2014a.

Felix Hill, Roi Reichart, and Anna Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *arXiv preprint arXiv:1408.3456*, 2014b.

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

Omer Levy and Yoav Goldberg. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 2, 2014a.

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

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations*, 2013a.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of Human Language Technologies: The 2013 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 746–751, 2013b.

George Miller and Walter Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28, 1991.

Andriy Mnih and Geoffrey Hinton. Three new graphical models for statistical language modelling. In *Proceedings of the 24th International Conference on Machine Learning*, pages 641–648. ACM, 2007.

Andriy Mnih and Geoffrey E Hinton. A scalable hierarchical distributed language model. In *Advances in Neural Information Processing Systems*, pages 1081–1088, 2009.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 95–100. Association for Computational Linguistics, 2012.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2014.

Richard Reichert, John Olney, and James Paris. *Two Dictionary Transcripts and Programs for Processing Them – The Encoding Scheme, Parsent and Conix.*, volume 1. DTIC Research Report AD0691098, 1969.

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

Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394, Uppsala, Sweden, July 2010.

Tong Wang and Graeme Hirst. Exploring patterns in dictionary definitions for synonym extraction. *Natural Language Engineering*, 17, 2011.

Yinggong Zhao, Shujian Huang, Xinyu Dai, Jianbing Zhang, and Jiajun Chen. Learning word embeddings from dependency relations. In *Proceedings of 2014 International Conference on Asian Language Processing (IALP)*, pages 123–127. IEEE, 2014.