Joint Unsupervised Learning of Semantic Representation of Words and Roles in Dependency Trees

Michal Konkol
NTIS – New Technologies for the Information Society,
Faculty of Applied Sciences, University of West Bohemia,
Technicka 8, 306 14 Plzen, Czech Republic
konkol@kiv.zcu.cz

Abstract

In this paper, we introduce WoRel, a model that jointly learns word embeddings and a semantic representation of word relations. The model learns from plain text sentences and their dependency parse trees. The word embeddings produced by WoRel outperform Skip-Gram and GloVe in word similarity and syntactical word analogy tasks and have comparable results on word relatedness and semantic word analogy tasks. We show that the semantic representation of relations enables us to express the meaning of phrases and is a promising research direction for semantics at the sentence level.

1 Introduction

Over the last few years, word level semantics was used with great success in many natural language processing tasks, e.g. named entity recognition (Lample et al., 2016), question answering (Yih et al., 2013), or sentiment analysis (Maas et al., 2011).

Skip-Gram (Mikolov et al., 2013a) and GloVe (Pennington et al., 2014) are among the most successful methods for word level semantics. Both methods are based on the Distributional Hypothesis (Harris, 1954), which says that words appearing in similar contexts have similar meaning. They represent semantics by dense high-dimensional vectors and words with similar vectors are supposed to have similar meaning. Levy et al. (2015) shows that Skip-Gram, GloVe, and some other methods can achieve similar results.

The semantics of higher level text units is currently one of the main research directions of natural language processing. A wide variety of algorithms was proposed, e.g. distributional tree kernels (Ferrone and Zanzotto, 2014), weighted combinations of word embeddings (Brychcín and Svoboda, 2016), neural networks (Socher et al., 2011; He et al., 2015), or extensions to word level semantics methods (Le and Mikolov, 2014).

A few authors extended Skip-Gram with dependency trees. Levy and Goldberg (2014a) redefine the context window to adjacent nodes in the dependency tree. Very similar approach to Levy and Goldberg (2014a) was used by Bansal et al. (2014) and Qiu et al. (2015). Bansal (2015) enhanced the context representation by adding new syntax-related features.

We propose a new method, called WoRel (Word Relations), that uses architecture similar to Skip-Gram, but learns not only word embeddings but also a representation of word relations that can be used to combine words into phrases. WoRel does not use the dependency trees to define syntax-based context as in the previous works, but tries to predict the context based on two words connected by an edge in the dependency tree.

2 Skip-Gram Model

Skip-Gram is a neural network model for word level semantics. It was introduced by Mikolov et al. (2013a). Later, Mikolov et al. (2013b) proposed a more efficient training procedure called negative sampling.

Skip-Gram represents each word by two d-dimensional vectors. We define a *middle word* as the word at the current position in the corpus and a *context word* as any word in a context window (a small neighborhood of the current position). The middle words are represented by vectors $\mathbf{m}_w \in \mathbf{R}$, where w is a word from the vocabulary $\mathbf{V}$. Context words are represented by vectors $\mathbf{c}_w \in \mathbf{R}^d$. We denote w_j the word at position j in the corpus. We maximize the negative sampling objective

Proceedings of Recent Advances in Natural Language Processing, pages 394–400,
Varna, Bulgaria, Sep 4–6 2017.
https://doi.org/10.26615/978-954-452-049-6_052

function

$$\sum_{\substack{k=j-l \\ k \neq j}}^{j+l} \log \sigma(\mathbf{m}_{w_j} \cdot \mathbf{c}_{w_k}) + \sum_{n \in \mathbf{N}} \log \sigma(-\mathbf{m}_{w_j} \cdot \mathbf{c}_n) \tag{1}$$

at each position j in the corpus. The size of the context l is selected randomly from 1 to L. $\mathbf{N}$ is a set of words (random samples) taken from a noise distribution, $\mathbf{N} = \{w \sim P_n(\mathbf{V})\}$.

3 WoRel Model

Skip-Gram uses a middle word (e.g. *food*) from a corpus to guess the words in the context window. If we know that another word (e.g. *rotten*) is related to the middle word, then we can use this information to improve our guess of the context (e.g. *excellent* becomes less probable).

WoRel does not use the middle word in the same way as Skip-Gram, but instead we have a pair of related words, called *phrase* from now on, represented by a vector $\mathbf{p}_j \in \mathbf{R}^d$, where j is the position in the corpus. We define related words as words that are connected by an edge in a dependency tree. A phrase at the position j consists of a word at the position j and its parent (head) in a dependency tree at the position $h(j)$ in the corpus. At position j in the corpus we maximize

$$\sum_{\substack{k=h(j)-l \\ k \notin \{j, h(j)\}}}^{h(j)+l} \log \sigma(\mathbf{p}_j \cdot \mathbf{c}_{w_k}) + \sum_{n \in \mathbf{N}} \log \sigma(-\mathbf{p}_j \cdot \mathbf{c}_n). \tag{2}$$

The phrase vector $\mathbf{p}_j$ is a function of the words w_j, $w_{h(j)}$, and their relation r_j in the dependency tree (e.g. subject or modifier):

$$\mathbf{p}_j = f(\mathbf{m}_{w_j}, \mathbf{m}_{w_{h(j)}}, r_j). \tag{3}$$

There are plenty of functions that can model the meaning of a phrase. We considered three options for the function – matrix multiplication, element-wise linear combination, and linear combination. On one hand, the model is trained on billions of tokens so the function cannot be too complex. On the other hand, too simple function may not be able to express the meaning of the phrase. We ended up with the element-wise linear combination (4) that seems to be a good trade-off between the speed and complexity.

$$\begin{aligned} &f(\mathbf{m}_{w_j}, \mathbf{m}_{w_{h(j)}}, r_j) \\ &\quad = \boldsymbol{\lambda}_{r_j} \odot \mathbf{m}_{w_{h(j)}} + (\mathbf{1} - \boldsymbol{\lambda}_{r_j}) \odot \mathbf{m}_{w_j} \end{aligned} \tag{4}$$

The vector $\boldsymbol{\lambda}_r \in [0,1]^d$ is a parameter vector for role r and acts as a filter for both words. The symbol $\odot$ denotes an element-wise multiplication.

The parameters of the model (vectors $\mathbf{c}_w$, $\mathbf{m}_w$, $\boldsymbol{\lambda}_r$ for all words w in the vocabulary and roles r) can be found using standard optimization methods, e.g. gradient descent.

4 Word Embeddings Experiments

4.1 Training Setup

We use a combination of the Gigaword corpus and the Wikipedia 2013 dump as the training data (approximately 2.5 billion words). The dependency trees are produced by Stanford neural network parser (Chen and Manning, 2014). The parser was chosen primarily for its speed. We chose universal dependencies parse trees (Nivre et al., 2016) because they can be used across languages and they place the semantically more important words closer to the root[1].

The model has several hyperparameters. They were set according to recommended values for Skip-Gram (Mikolov et al., 2013b; Levy and Goldberg, 2014a). We use maximum context size $L = 10$, number of negative samples $|\mathbf{N}| = 10$, learning rate $\alpha = 0.025$, dimension of semantic vectors $d = 300$, vocabulary size $|\mathbf{V}| = 300\,000$, unigram word distribution raised to 0.75 as the negative sample distribution $P_n(\mathbf{V})$. We do not use subsampling in WoRel and do not remove rare words (it would corrupt the parse trees).

4.2 Evaluation

We evaluate WoRel on two standard tasks: word similarity and word analogy. In evaluation we represent each word with vector $\mathbf{v}_w = \mathbf{m}_w + \mathbf{c}_w$ the same way as in GloVe.

Word similarity. The word similarity and relatedness corpora consists of word pairs and their similarity scores assigned by human annotators. The goal of the algorithm is to assign scores that maximize Spearman correlation

[1]See examples of preposition and conjunction roles at `http://universaldependencies.org`

	RG	WordSim			Google Word Analogy		
		all	rel	sim	all	syn	sem
Skip-Gram – recommended †	–	–	.623	.773	.599	–	–
Skip-Gram – tuned †	–	–	.700	.794	.694	–	–
GloVe – tuned †	–	–	**.746**	.643	.702	–	–
Skip-Gram – LS †	–	–	.681	.766	**.739**	–	–
GloVe – LS †	–	–	.624	.678	.732	–	–
Skip-Gram ‡	.628	.697	–	–	.691	.660	.730
GloVe ‡	.778	.658	–	–	.717	.670	**.774**
Skip-Gram – BoW 5 §	.776	.686	.607	.751	.613	.615	.610
Skip-Gram – BoW 2 §	.727	.657	.567	.737	.539	.627	.532
Skip-Gram – dependency §	.771	.626	.492	.754	.361	.526	.162
WoRel	**.817**	**.733**	.685	**.803**	.731	**.727**	.735

Table 1: Results of WoRel compared with other methods on the word similarity datasets WordSim-353 and RG-65 and the Google Word Analogy dataset. † Results from (Levy et al., 2015). ‡ Results from (Pennington et al., 2014). § Embeddings provided by Levy and Goldberg (2014a).

with the annotated scores. We use Rubenstein-Goodenough corpus (Rubenstein and Goodenough, 1965), WordSim-353 corpus (Finkelstein et al., 2001), and WordSim-353 partitioned to similarity and relatedness corpora (Agirre et al., 2009).

Word analogy. In the word analogy task the model answers questions in the form "What word (d) is related to c in the same way as b is related to a?" E.g. if a is France, b is Paris, and c is Germany we would expect d to be Berlin. The quality of the model is measured by accuracy. We use the Google Word Analogy corpus and its semantic and syntactic partitions. We do not remove questions with out-of-vocabulary words as in (Levy and Goldberg, 2014a) because it favors smaller vocabularies. In our experiments we use the original equation (5) to choose word d, where $\cos\mathrm{sim}(x, y)$ denotes the cosine similarity between x and y. Even though the 3CosMul approach (Levy and Goldberg, 2014b) has better results we use the older approach for a fair comparison with previous works.

$$d = \operatorname*{arg\,max}_{w \in \mathbf{V} \setminus \{a,b,c\}} \cos\mathrm{sim}(\mathbf{v}_b - \mathbf{v}_a + \mathbf{v}_c, \mathbf{v}_w) \quad (5)$$

4.3 Results and Discussion

The results for word similarity and analogy tasks are in Table 1 together with previously published results of Skip-Gram and GloVe. We present several results from (Levy et al., 2015). We start with Skip-Gram with the *recommended* hyperparameters. This configuration is used in most cases. The models denoted by *tuned* use hyperparameters that were found using cross-validation. This approach is not usable in most cases as it requires supervision and it would be too demanding to set all hyperparameters for all tasks to optimal values. But it gives us an upper limit to expected results. The models denoted *LS* use much bigger data than the previous models (10.5 billion words, previous models 1.5 billion tokens) and the hyperparameters are also tuned using cross-validation, but less combinations were tested due to longer training times.

We also compare our results with (Pennington et al., 2014). They provide results for Skip-Gram and GloVe trained on a corpus with 6 billion tokens.

The last comparison is with Skip-Gram with dependency (and also bag-of-word) contexts provided by Levy and Goldberg (2014a). We see that the dependency Skip-Gram is significantly worse than WoRel or even other models. Levy and Goldberg (2014a) showed that their model is very good for different purposes (e.g. classification between relatedness and similarity).

The results show the strengths of WoRel. It significantly outperforms Skip-Gram and GloVe on the syntactical word analogies (5-6% in absolute values). If we consider that we use dependency trees during the training, it may not be so surprising. More surprising are Worel's excellent results on tasks that focus on word similarity (in contrast to relatedness) – RG-65 and WordSim-353 similarity partition. We believe that this is because the similarity is connected to syntax much more than relatedness and WoRel is better at modeling syntax.

Target Phrase	Skip-Gram BoW 5	Skip-Gram Dependency	WoRel
police officer	officers	policeman	policeman
	lapd	officers	sergeant
	inspector	patrolman	constable
	sergeant	síochána	officers
	plainclothes	gardaí	inspector
army officer	corps	artilleryman	sergeant
	commander	brigadeführer	commander
	commandant	nco	soldier
	quartermaster	signaller	colonel
	commanding	militiaman	lieutenant
life partner	mentor	archnemesis	partners
	friend	protegee	girlfriend
	colleague	coworker	friend
	roommate	step-sister	colleague
	partners	love-interest	collaborator
business partner	partners	buisness	partners
	firm	distributorship	firm
	stockbroking	stockholder	shareholder
	partnership	sub-contractor	investor
	import-export	syndicator	supplier
scientific publication	scholarly	bibliographical	periodical
	periodical	musicological	journal
	publications	newpaper	publishing
	journals	journalistic	publications
	triannual	ezine	periodicals
snow falls	rain	snows	snowfall
	sleet	sleet	rain
	snows	thaws	snows
	lake-effect	snowdrifts	snowfalls
	helmcken	rain	rains
make decision	decisions	”to	decide
	overrule	withdrawl	bring
	overturn	kowtow	overturn
	making	forbear	impose
	second-guess	ceteris	give
provide proof	providing	theorise	demonstrate
	substantiation	impute	prove
	verification	proove	give
	provides	substantiation	make
	demonstrate	adduce	satisfy

Table 2: Five best replacements for a target phrase provided by WoRel and Skip-Gram with bag-of-word and dependency contexts.

5 Phrase Embeddings Experiments

We believe the representation of word relations is the most innovative and promising part of WoRel. In this section we use them to create phrase embeddings and provide a qualitative and quantitative analysis of these embeddings. We are not aware of any existing corpora that could be easily used for a standard quantitative analysis of word relation representations thus we propose our own evaluation.

Our experiments with word relations are based on (non-idiomatic) phrases that can be expressed by a single word with a similar meaning. WoRel is compared with two baselines: Skip-Gram embeddings with bag-of-word and dependency contexts provided by Levy and Goldberg (2014a). The phrase embeddings are obtained using Equation (4) for WoRel and an unweighted linear combination of the words for the Skip-Gram baselines, a common approach for phrase representation with Skip-Gram (Agirre et al., 2016).

In Table 2 we show five most similar words for a few target phrases. We believe that the provided examples show the quality of WoRel phrase embeddings and that WoRel is able to choose better replacements for the target phrases. We believe that the improvement comes from the WoRel cost function which directly requires a phrase embedding in the same space as word embeddings.

We use the same approach in quantitative analysis. Firstly, we selected a set of phrases that have multiple single-word equivalents. The set was filtered to contain only word phrases where the models differ significantly ($\approx$ 20–30%) in order to reduce annotator work. To avoid author bias we filtered the phrases blindly, i.e. the models were listed in random order. The final set contains 20 phrases. For each phrase the models were ranked blindly by four annotators. Ties (e.g. rankings 1-1-3, 1-1-1, 1-2-2) were allowed. The average standard deviation of the assigned rank is 0.4.

The results of this experiment are in Table 3. For each model we show the sum of all the ranks assigned by individual annotators and the overall results. The best achievable result is 20 for individual annotators and 80 overall (the model is the best for all phrases). The worst result is 60 for individual annotators and 240 overall (third place for all phrases). We can see that WoRel (average rank 1.29) significantly outperforms both baselines (average ranks 2.14 and 2.31).

For a better idea of the relation representations we provide a visualization of a few common dependency roles on Figure 1. By observing some patterns (a few of them circled) in the representations we see that the model learns that the role `nsubj` (subject) is very similar to role `nsubjpass` (passive subject) and in both roles there is almost equal importance of child (usually noun) and parent (usually verb). The roles `det` (determiner) and `amod` ($\approx$ adjective) are in some aspects similar to each other. For these roles the child words have smaller semantic importance than the parent.

Model	Annotator 1	Annotator 2	Annotator 3	Annotator 4	Total	Average rank
Skip-Gram Dependency	44	43	41	43	171	2.14
Skip-Gram BoW 5	45	51	41	41	185	2.31
WoRel	22	26	27	28	103	1.29

Table 3: The sum of ranks assigned by annotators. Lower numbers are better.

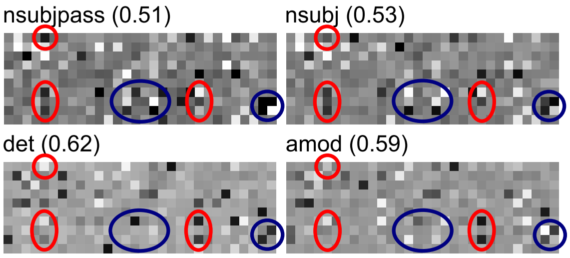

Figure 1: Role representations for selected universal dependency roles. The darker (lighter) the color is, the more information comes from the child (parent). Values in parenthesis show overall importance of the parent (average value of λ_r).

6 Conclusion and Future Work

We proposed WoRel, a new distributional semantics model based on Skip-Gram. The main contribution of WoRel is that it learns not only word embeddings, but also the representations of dependency relations between words.

The word embeddings were tested on word similarity and analogy tasks. WoRel significantly outperformed Skip-Gram and GloVe on syntactical word analogy and word similarity tasks and had similar results to Skip-Gram on semantic word analogy and word relatedness tasks.

Even though the improvement in word embeddings is important, the main innovation lies in the representation of word relations. The relation representations have interesting semantic properties and can be used in a variety of NLP tasks. More importantly, we believe that they can be used to represent semantics at the sentence level.

Our further research will focus on the semantic representation of sentences. WoRel is able to represent meaning of a single edge in a dependency tree (combine a child with its parent), but it is necessary to find a way to properly combine edges with a common parent and ensure transition of the semantic information from leaves of the dependency tree to the root.

Other directions for further research include evaluation on several NLP tasks, finding the optimal hyperparameters of the model, exploring the effect of data size, proposing other representations of the context (e.g. dependency), or employing more robust and efficient methods for optimization.

The reference implementation and trained word embeddings are publicly available at the authors web pages[2].

Acknowledgments

This publication was supported by the project LO1506 of the Czech Ministry of Education, Youth and Sports under the program NPU I.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, NAACL '09, pages 19–27.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, pages 497–511. http://www.aclweb.org/anthology/S16-1081.

Mohit Bansal. 2015. Dependency link embeddings: Continuous representations of syntactic substructures. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*. Association for Computational Linguistics, Denver, Colorado, pages 102–108.

Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2014. Tailoring continuous word representations for dependency parsing. In *Proceedings of the 52nd*

[2]See `http://konkol.me/publications/Konkol-RANLP_2017.html`

Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Association for Computational Linguistics, Baltimore, Maryland, pages 809–815.

Tomáš Brychcín and Lukáš Svoboda. 2016. Uwb at semeval-2016 task 1: Semantic textual similarity using lexical, syntactic, and semantic information. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, pages 588–594.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Lorenzo Ferrone and Fabio Massimo Zanzotto. 2014. Towards syntax-aware compositional distributional semantic models. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin City University and Association for Computational Linguistics, Dublin, Ireland, pages 721–730.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th International Conference on World Wide Web*. ACM, New York, NY, USA, WWW '01, pages 406–414. https://doi.org/10.1145/371920.372094.

Zellig S. Harris. 1954. Distributional structure. *Word* .

Hua He, Kevin Gimpel, and Jimmy Lin. 2015. Multi-perspective sentence similarity modeling with convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 1576–1586.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 260–270.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In Tony Jebara and Eric P. Xing, editors, *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. JMLR Workshop and Conference Proceedings, pages 1188–1196.

Omer Levy and Yoav Goldberg. 2014a. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Baltimore, Maryland, pages 302–308.

Omer Levy and Yoav Goldberg. 2014b. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Ann Arbor, Michigan, pages 171–180. http://www.aclweb.org/anthology/W14-1618.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *TACL* 3:211–225.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '11, pages 142–150.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*. pages 3111–3119.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association, Paris, France, pages 1659–1666.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543.

Likun Qiu, Yue Zhang, and Yanan Lu. 2015. Syntactic dependencies and distributed word representations for analogy detection and mining. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 2441–2450.

Herbert Rubenstein and John B. Goodenough. 1965. Contextual correlates of synonymy. *Commun. ACM* 8(10):627–633. https://doi.org/10.1145/365628.365657.

Richard Socher, Eric H. Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Y. Ng. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, Curran Associates, Inc., pages 801–809.

Wen-tau Yih, Ming-Wei Chang, Christopher Meek, and Andrzej Pastusiak. 2013. Question answering using enhanced lexical semantic models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Sofia, Bulgaria, pages 1744–1753.