Incorporating Contextual and Syntactic Structures Improves Semantic Similarity Modeling

Linqing Liu,[1] **Wei Yang,**[1] **Jinfeng Rao,**[2] **Raphael Tang,**[1] and **Jimmy Lin**[1]
[1] David R. Cheriton School of Computer Science, University of Waterloo
[2] Department of Computer Science, University of Maryland, College Park
{linqing.liu, w85yang, r33tang, jimmylin}@uwaterloo.ca, raojinfeng@gmail.com

Abstract

Semantic similarity modeling is central to many NLP problems such as natural language inference and question answering. Syntactic structures interact closely with semantics in learning compositional representations and alleviating long-range dependency issues. However, such structure priors have not been well exploited in previous work for semantic modeling. To examine their effectiveness, we start with the Pairwise Word Interaction Model, one of the best models according to a recent reproducibility study, then introduce components for modeling context and structure using multi-layer BiLSTMs and TreeLSTMs. In addition, we introduce residual connections to the deep convolutional neural network component of the model. Extensive evaluations on eight benchmark datasets show that incorporating structural information contributes to consistent improvements over strong baselines.

1 Introduction

Modeling the semantic similarity between a pair of sentences is a fundamental task in natural language processing. It is the core problem of many tasks such as question answering (He et al., 2015; Rao et al., 2017; Wang et al., 2018) and query ranking (Mitra and Craswell, 2019). Recently, various neural networks have been proposed for textual similarity modeling. These models share three main components: (1) sequential sentence encoders, which incorporate word context and sentence order for better sentence representations, e.g., by using recurrent neural networks (RNNs; Mikolov et al., 2010; Seo et al., 2016), (2) interaction and attention mechanisms, which use the encoding outputs of sentences to calculate or reweight salient word pair interactions (He and Lin, 2016; Chen et al., 2017), and (3) incorporating syntactic parsing information as an intuitive structure prior for sentence modeling (Chen and Manning, 2014; Zhao et al., 2016; Chen et al., 2017).

Our work is inspired by the recent reproducibility study by Lan and Xu (2018), which examines many neural network architectures for semantic similarity modeling through extensive evaluations on multiple benchmark datasets. Their results suggest that syntactic structure information captured by a TreeLSTM encoder either provides few benefits or even hurts performance. Structure information has often been overlooked in recent semantic modeling methods, such as InferSent (Conneau et al., 2017), DecAtt (Parikh et al., 2016), and BiMPM (Wang et al., 2017). It is not yet clear whether the syntactic structures implicitly captured by sequential modeling of texts from large annotated data or existing structure modeling techniques (Tai et al., 2015; Kipf and Welling, 2017) are effective in learning tree representations.

To further explore the effects of tree structures in sentence modeling, we start with the Pairwise Word Interaction Model (PWIM) of He and Lin (2016) as our base architecture, which has shown strong performance on various datasets from Lan and Xu (2018). In summary, PWIM uses a BiLSTM to learn word-level context vectors from both input sentences and builds a novel similarity focus layer with pairwise metrics to identify important word pairs. It then converts the similarity measurement problem to a pattern recognition problem for the final classification. We argue that PWIM approaches semantic modeling from a word-level matching perspective, and hence fails to capture syntactic and contextual semantics. To this end, we add multi-layer BiLSTMs with shortcut connections to capture long-range context, as well as TreeLSTM encoders to capture the syntactic structure of sentences.

We conduct thorough evaluations across eight

datasets in four NLP tasks: paraphrase identification, semantic textual similarity, natural language inference, and answer sentence selection. Extensive experiments show that our proposed components lead to consistent improvements against PWIM and other strong baselines, suggesting that incorporating contextual and syntactic structures can help semantic modeling. Our improved model achieves competitive numbers on eight datasets, and we open-source our code to improve reproducibility and to facilitate future research.[1]

2 Methods

2.1 The Pairwise Word Interaction Model

The Pairwise Word Interaction model (He and Lin, 2016) captures fine-grained word-level information to measure textual similarity. They use a Bi-LSTM for context modeling, where the word at time step t is encoded as a forward hidden state h_t^{for} and a backward hidden state h_t^{back}. Pairwise word interactions are modeled through a multi-metric comparison unit coU, which computes cosine distance, L_2 distance, and dot-product distance over two hidden states. This comparison unit is applied to not only the forward and backward hidden states h_t^{for} and h_t^{back}, but also their concatenation $\overleftrightarrow{h_t} = [h_t^{\text{for}}, h_t^{\text{back}}]$ and summation $h_t^{+} = h_t^{\text{for}} + h_t^{\text{back}}$.

The output is a similarity tensor of size $R^{13\times|sent1|\times|sent2|}$ with one extra dimension for the padding indicator. Instead of using attention weight vectors or weighted representations, He and Lin apply a focus layer on the similarity tensor to decrease the weights of unimportant word interactions by a factor of ten. They then consider the tensor as an "image" with 13 channels and use a 19-layer-deep convolutional neural network to predict the final classification.

2.2 Residual Connections

Since He and Lin (2016) phrase the similarity measurement problem as a pattern recognition (image processing) problem and apply deep convolutional neural networks, we explore the addition of residual connections (He et al., 2016) to deal with the potential vanishing gradient problems in deep networks. A building block is defined as $y = f(x, W_i) + x$, where x, y are the input and output of the layer considered, and $f(x, W_i)$ is the learned residual mapping.

[1] https://github.com/likicode/spwim

2.3 Multi-layer BiLSTM Sentence Encoders

We use multiple stacked, bi-directional LSTM layers with shortcut connections, similar to Nie and Bansal (2017). In this architecture, the input sequences of the i^{th} BiLSTM layer are the concatenated outputs of all the previous layers and the initial word embedding sequences. Let $W = \{w_1, w_2, ..., w_n\}$ represent the word embeddings associated with each word in the source sentence. Define the output of the i^{th} BiLSTM layer at time t as $h_t^i = \text{BiLSTM}^i(x_t^i)$. Then, the input of the i^{th} BiLSTM layer at time t is:

$$x_t^1 = w_t, \;\; x_t^i = [w_t, h_t^1, ..., h_t^{i-2}, h_t^{i-1}](i > 1)$$

Differing from the original paper, we directly use the output of the last BiLSTM layer to encode the sentence $v = (h_1^m, h_2^m, ...h_n^m)$, where n denotes the length of the sentence and m the number of BiLSTM layers. In our experiments we set $m = 3$.

2.4 Hybrid Inference Model with Parse Trees

While stacked BiLSTMs capture long-term dependency and contextual information over each sentence, we are also interested in investigating explicit hierarchical relationships among linguistic phrases and clauses. To incorporate this domain-specific information, we use the Dependency TreeLSTM (Tai et al., 2015), whose nodes condition their components on the sum of the hidden states of their children. Suppose h_L and h_R are the sentence representations in the pair over the parse tree of each sentence: to model similarity, we compute the element-wise product $h_L \odot h_R$ and absolute difference $|h_L - h_R|$. Then, we feed the two similarity vectors to a fully-connected layer with softmax whose output is the probability distribution over labels. To compute the final label for the sentence pair, we interpolate between the output probabilities of this model and those of PWIM. Chen et al. (2017) also incorporate tree structures produced by a constituency parser into the ESIM model, then average the predicted probabilities.

3 Experimental Setup

We conducted experiments on eight separate datasets—one natural language inference dataset, two paraphrase identification datasets, three SemEval competition datasets, and two QA datasets—which are as follows:

- SNLI (Bowman et al., 2015) is a collection of 570k manually-labeled sentence pairs for

Dataset	SNLI	Quora	Twitter	PIT-2015	STS-2014	WikiQA	TrecQA	SICK
	Acc	Acc	F1	F1	Pearson's r	MAP/MRR	MAP/MRR	Pearson's r/ρ
InferSent	0.846	0.866	0.746	0.451	0.715	0.287/0.287	0.521/0.559	-
SSE	0.855	0.878	0.650	0.422	0.378	0.624/0.638	0.628/0.670	-
DecAtt	0.856	0.845	0.652	0.430	0.317	0.603/0.619	0.660/0.712	-
ESIM$_{tree}$	0.864	0.755	0.740	0.447	0.493	0.618/0.633	0.698/0.734	-
ESIM$_{seq}$	0.870	0.850	0.748	0.520	0.602	0.652/0.664	0.771/0.795	-
ESIM$_{seq+tree}$	0.871	0.854	0.759	0.538	0.589	0.647/0.658	0.749/0.768	-
PWIM$_{our}$	0.822	0.853	0.745	0.602	0.695	0.709/0.723	0.759/0.822	0.871/0.809
mPWIM$_{seq}$	0.851	0.862	0.757	0.612	0.714	0.717/0.728	0.774/**0.835**	0.878/0.821
mPWIM$_{seq+tree}$	0.855	0.870	0.743	**0.623**	**0.718**	**0.735/0.751**	**0.781**/0.821	**0.887/0.834**
Abs increase (%)	3.3	1.7	-	2.1	2.3	2.6/2.8	2.2/-	1.6/2.5

Table 1: Test results on different datasets.

the task of natural language inference. The relationship between two sentences includes entailment, contradiction, and neutral.

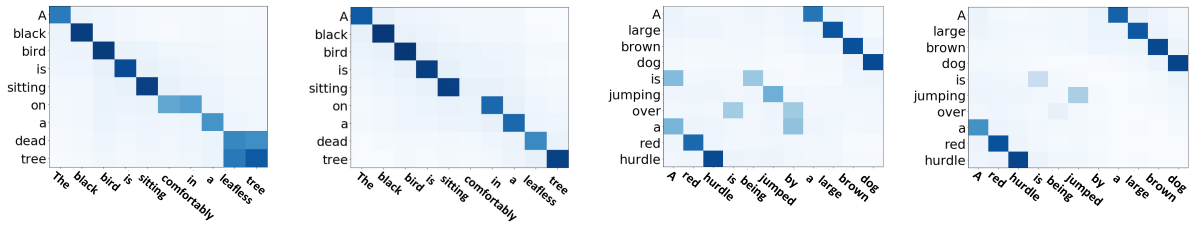
- Quora (Iyer et al., 2017) consists of 400k question pairs collected from the Quora website, with binary labels indicating if they are duplicates of each other.
- Twitter-URL (Lan et al., 2017) is a paraphrase corpus with 50k sentence pairs.
- PIT-2015 (Xu et al., 2015) is a paraphrase dataset that comes from SemEval-2015 Task 1.
- STS-2014 (Agirre et al., 2014) comes from SemEval-2014 Task 10 and each pair of sentences has a similarity score $\in [0, 5]$.
- WikiQA (Yang et al., 2015) is an open-domain question-answering dataset. After applying the same pre-processing methods in He and Lin (2016), it contains 12k question-answer pairs with binary labels.
- TrecQA (Wang et al., 2007) is from the Text Retrieval Conferences and consists of 56k question-answer pairs.
- SICK (Marelli et al., 2014) comes from SemEval-2014 Task 1 with 10k annotated sentence pairs. Each pair has a similarity score $\in [1, 5]$.

The first seven datasets are the same as the ones examined in Lan and Xu (2018), except for MNLI (Williams et al., 2018), since SNLI is much larger than MNLI for the task of natural language inference. We also add the SICK dataset (Marelli et al., 2014), which is unexplored in Lan and Xu (2018). Across multiple tasks and domains, we systematically compare our proposed models with state-of-the-art neural models: InferSent (Conneau et al., 2017), Shortcut-stacked Sentence Encoder (SSE; Nie and Bansal, 2017), Decomposable Attention Model (DecAtt; Parikh et al., 2016), and Enhanced Sequential Inference Model (ESIM; Chen et al., 2017).

For our experiments on SNLI, Quora, Twitter-URL, PIT-2015, WikiQA and TrecQA, the training objective is to minimize the NLL loss. For STS-2014 and SICK datasets, we use the KL divergence loss. Following He and Lin (2016), for all cases, we use the RMSProp optimizer (Tieleman and Hinton, 2012). Our word representations use 300-dimensional GloVe word vectors (Pennington et al., 2014), which we make static in all experiments. We produce dependency parse trees for each sentence using the Stanford Neural Network Dependency Parser (Chen and Manning, 2014). The TreeLSTM then models sentence representations over each sentence's parse tree.

4 Results and Analysis

Table 1 shows the results of our models on different datasets. The first block of the table contains figures copied directly from Lan and Xu (2018); note that they do not use SICK. PWIM$_{our}$ refers to our own implementation. Note that there are at least three independent open-source implementations of the PWIM base model that we are aware of, which confirms the robustness and reproducibility of the model. Most results from these implementations are consistent; however, for



(a) mPWIM$_{seq}$ (b) mPWIM$_{seq+tree}$ (c) mPWIM$_{seq}$ (d) mPWIM$_{seq+tree}$

Figure 1: Visualization of cosine values in the focusCube of two sentence pairs in the SICK test set.

PIT-2015 and STS-2014, we observe some differences, which we were unable to reconcile even after contacting the previous authors. Thus, for comparison purposes, we report results from our base PWIM$_{our}$ implementation.

4.1 Effects of the Multi-Layer BiLSTM

The entry mPWIM$_{seq}$ denotes PWIM using multi-layer BiLSTMs for modeling the context of the input sentences and also incorporating residual connections in the final classification. On all datasets listed in the table, adding multi-layer BiLSTMs leads to a higher performance than that of the original model PWIM$_{our}$.

SSE (Nie and Bansal, 2017) is a stacked BiLSTM model with shortcut connections and fine-tuning of word embeddings. Unlike our setting, where each word is represented by its own hidden state in the final output layer, SSE applies max-pooling over time to the output of the last BiLSTM layer to extract the final sentence feature vector. Based on Table 1, mPWIM$_{seq}$ clearly outperforms SSE on Twitter, PIT-2015, STS-2014, WikiQA, and TrecQA. However, for the SNLI and Quora datasets, SSE slightly exceeds mPWIM by 0.4% and 1.6%, respectively. SNLI and Quora have the largest training data among all the datasets with 550k and 393k training sentence pairs, respectively, which suggests that SSE performs better on larger data beyond a certain threshold. We surmise that as the dataset increases in size, the simplicity of SSE will have more performance advantages.

4.2 Effects of TreeLSTM

The mPWIM$_{seq+tree}$ further enhances mPWIM$_{seq}$ by incorporating syntactic TreeLSTMs based on syntactic parse trees of each sentence. It averages the prediction probabilities of the PWIM using multi-layer BiLSTMs and TreeLSTMs separately to arrive at the final label. ESIM$_{seq+tree}$ also computes its final predictions by averaging prediction probabilities of two ESIM variants that use BiLSTMs and TreeLSTMs as sentence encoders, respectively.

From the table, we observe that adding TreeLSTMs to the ESIM model only marginally helps or has no effect for most datasets. On the other hand, TreeLSTM complements PWIM well: for WikiQA, it increases mean average precision (MAP) by 1.8% and mean reciprocal rank (MRR) by 2.3%. TreeLSTM also contributes to an 1.1% increase in the F1-measure for PIT-2015, 0.9% Pearson's r for SICK, and 0.7% MAP for TrecQA.

We hypothesize that these observed differences can be attributed to the model architectures. The inference model of ESIM is based on chain LSTMs, which might encode overlapping information with TreeLSTMs. For PWIM, the sentence context information is transformed into pairwise word interaction similarity units, and then a 19-layer-deep CNN exploits the spatially localized patterns. During this process, its focus is word-level similarities in sentences. The syntactic parsing structure introduced by TreeLSTM compensates for some of the information deficiencies. Notably, TreeLSTM does not help PWIM on the Twitter dataset; this makes sense, as Kong et al. (2014) note that many elements in tweets have no syntactic function, including hashtags and URLs. Furthermore, tweets often contain multiple fragments, each with its own syntactic span. Both of these issues may degrade the quality of the syntactic modeling of tweets.

4.3 Sample Visualization and Analysis

To better understand *why* our models achieve improved effectiveness, we visualize the cosine values of the focusCube (the final output of the similarity layer) for pairwise word interactions in mPWIM$_{seq}$ and mPWIM$_{seq+tree}$, using the same method as Chen et al. (2017), where darker colors indicate stronger pairwise word interactions.

In Figure 1, we show visualizations from two pairs of sentences from SICK: 1a and 1b form a contrastive pair, as do 1c and 1d. We see that, in both cases, the TreeLSTM helps the model find syntactically important pairwise word interactions. For example, in Figure 1a, for the mPWIM$_{seq}$ model, the cluster of dark patches near the top shows obviously irrelevant correspondences, e.g., "on" with "comfortably", "dead" with "tree", and several of the articles are misaligned with respect to their positions in phrase structure. With the incorporation of syntactic information in Figure 1b, the correspondences are much more accurate.

We see that this is similarly the case when comparing Figures 1c and 1d, where the TreeLSTM yields more accurate correspondences. With the TreeLSTM, the model has learned the correct correspondence between "is being jumped" and "is jumping over", whereas without the syntactic structure, the correspondences are quite muddled. In both cases, we observe that mPWIM$_{seq+tree}$ is able to capture the passive construction for paraphrase detection.

5 Conclusion

We examine the hypothesis of whether incorporating contextual and syntactic structures can improve semantic similarity modeling. We extend the strong PWIM model and add additional components comprised of TreeLSTMs and multi-layer BiLSTMs to capture syntax and context information. Thorough experiments on eight datasets show that our improved models achieve consistent gains in effectiveness.

Acknowledgments

This research was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada, and enabled by computational resources provided by Compute Ontario and Compute Canada.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 task 10: multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680.

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*, pages 1576–1586, Lisbon, Portugal.

Hua He and Jimmy Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 937–948.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. *First Quora Dataset Release: Question Pairs*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations*.

Lingpeng Kong, Nathan Schneider, Swabha Swayamdipta, Archna Bhatia, Chris Dyer, and Noah A. Smith. 2014. A dependency parser for tweets. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1001–1012.

Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. 2017. A continuously growing dataset of sentential paraphrases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1224–1234.

Wuwei Lan and Wei Xu. 2018. Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3890–3902.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. SemEval-2014 task 1: evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 1–8.

Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1045–1048.

Bhaskar Mitra and Nick Craswell. 2019. An introduction to neural information retrieval. *Foundations and Trends in Information Retrieval*, 13(1):1–126.

Yixin Nie and Mohit Bansal. 2017. Shortcut-stacked sentence encoders for multi-domain inference. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, pages 41–45.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Jinfeng Rao, Hua He, and Jimmy Lin. 2017. Experiments with convolutional neural network models for answer selection. In *Proceedings of the 40th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017)*, pages 1217–1220, Tokyo, Japan.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv:1611.01603*.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566.

Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5-RMSProp: divide the gradient by a running average of its recent magnitude. *Coursera: Neural Networks for Machine Learning*, 4(2):26–31.

Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. What is the Jeopardy model? A quasi-synchronous grammar for QA. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 22–32.

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerry Tesauro, Bowen Zhou, and Jing Jiang. 2018. R^3: Reinforced ranker-reader for open-domain question answering. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 5981–5988.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4144–4150.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122.

Wei Xu, Chris Callison-Burch, and Bill Dolan. 2015. SemEval-2015 task 1: paraphrase and semantic similarity in Twitter (PIT). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 1–11.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: a challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018.

Kai Zhao, Liang Huang, and Mingbo Ma. 2016. Textual entailment with structured attentions and composition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2248–2258.