An Empirical Study on End-to-End Sentence Modelling

Kurt Junshean Espinosa
National Centre for Text Mining, School of Computer Science, University of Manchester, UK
Department of Computer Science, University of the Philippines Cebu, Philippines
kurtjunshean.espinosa@postgrad.manchester.ac.uk
kpespinosa@up.edu.ph

Abstract

Accurately representing the meaning of a piece of text, otherwise known as sentence modelling, is an important component in many natural language inference tasks. We survey the spectrum of these methods, which lie along two dimensions: input representation granularity and composition model complexity. Using this framework, we reveal in our quantitative and qualitative experiments the limitations of the current state-of-the-art model in the context of sentence similarity tasks.

1 Introduction

Accurately representing the meaning of a piece of text remains an open problem. To illustrate why it is difficult, consider the pair of sentences *A* and *B* below in the context of a sentence similarity task.

```
A:The shares of the company dropped.
B:The organisation's stocks slumped.
```

If we use a very naïve model such as bag-of-words to represent a sentence and use discrete counting of common words between the two sentences to determine their similarity, the score would be very low although they are highly similar. How then do we represent the meaning of sentences?

Firstly, we must be able to represent them in ways that computers can understand. Based on the Principle of Compositionality (Frege, 1892), we define the meaning of a sentence as a function of the meaning of its constituents (i.e., words, phrases, morphemes). Generally, there are two main approaches to representing constituents: localist and distributed representations. With the localist representation[1], we represent each constituent with a unique representation usually taken from its position in a vocabulary *V*. However, this kind of representation suffers from the curse of dimensionality and does not consider the syntactic relationship of a constituent with other constituents. These two shortcomings are addressed by the distributed representation (Hinton, 1984) which encodes a constituent based on its co-occurrence with other constituents appearing within its context, into a dense *n*-dimensional vector where $n \ll |V|$. Estimating the distributed representation has been an active research topic in itself. Baroni et al. (2014) conducted a systematic comparative evaluation of context-counting and context-predicting models for generating distributed representations and concluded that the latter outperforms the former, but Levy et al. (2015) later have shown that simple pointwise mutual information (PMI) methods also perform similarly if they are properly tuned. To date, the most popular architectures to efficiently estimate these distributed representations are *word2vec* (Mikolov et al., 2013a) and *GloVe* (Pennington et al., 2014). Subsequent developments estimate distributed representations at other levels of granularity (see Section 2.1).

While much research has been directed into constructing representations for constituents, there has been far less consensus regarding the representation of larger semantic structures such as phrases and sentences (Blacoe and Lapata, 2012). A simple approach is based on looking up the vector representation of the constituents (i.e., embeddings) and taking their sum or average which yields a single vector of the same dimension. This strategy is effective in simple tasks but loses word order information and syntactic relations in the process (Mitchell and Lapata, 2008; Turney et al., 2010). Most modern neural network models have a sentence encoder that learns the representation of sentences more efficiently while preserving word or-

[1] The best example of this sparse representation is the "one-hot" representation (see Appendix A for details)

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics- Student Research Workshop, pages 128–135
Vancouver, Canada, July 30 - August 4, 2017. ©2017 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17-3021

der and compositionality (see Section 2.1).

In this work, we present a generalised framework for sentence modelling based on a survey of state-of-the-art methods. Using the framework as a guide, we conducted preliminary experiments by implementing an end-to-end version of the state-of-the-art model in which we reveal its limitations after evaluation on sentence similarity tasks.

2 Related Work

The best way to evaluate sentence models is to assess how they perform on actual natural language inference (NLI) tasks. In this work, we examine three related tasks which are central to natural language understanding: paraphrase detection (Dolan et al., 2004; Xu et al., 2015), semantic similarity measurement (Marelli et al., 2014; Xu et al., 2015; Agirre et al., 2016a) and interpretable semantic similarity measurement (Agirre et al., 2016b). (We refer the reader to the respective papers for the task description and dataset details).

Among the four broad types of methods we have identified in the literature (see Appendix C.1), we focus in this paper on deep learning (DL) methods because they support end-to-end learning, i.e., they use few hand-crafted features—or none at all, making them easier to adapt to new domains. More importantly, these methods have obtained comparable performance relative to other top-ranking methods.

2.1 Sentence Modelling Framework

As a contribution of this work, we survey the spectrum of DL methods, which lie on two dimensions: input representation granularity and composition model complexity, which are both central to sentence modelling (see Appendix Figure C.2 for a graphical illustration).

The first dimension (see horizontal axis of Appendix Figure C.2) is the granularity of input representation. This dimension characterises a tradeoff between syntactic dependencies captured in the representation and data sparsity. On the one hand, character-based methods (Vosoughi et al., 2016; dos Santos and Zadrozny, 2014; Wieting et al., 2016) are not faced with the data sparsity problem; however, it is not straightforward to determine whether composing sentences based on individual character representations would represent the originally intended semantics. On the other hand, while sentence embeddings (Kiros et al., 2015), which are learned by predicting the previous and next sentences given the current sentence, could intuitively represent the actual semantics, it suffers from data sparsity.

The second dimension (see vertical axis of Appendix Figure C.2) is the spectrum of composition models ranging from bag-of-items-driven[2] architectures to compositionality-driven ones to account for the morphological, lexical, syntactic, and compositional aspects of a sentence. Some of the popular methods are based on Bag-of-Item models, which represent a sentence by performing algebraic operations (e.g., addition or averaging) over the vector representations of individual constituents (Blacoe and Lapata, 2012). However, these models have received criticism as they use linear bag-of-words context and thus do not take into account syntax. *Spatial neural networks*, e.g., Convolutional Neural Networks or ConvNets (LeCun et al., 1998), have been shown to capture morphological variations in short subsequences (dos Santos and Zadrozny, 2014; Chiu and Nichols, 2016). However, this architecture still does not capture the overall syntactic information. Thus Sutskever et al. (2014) proposed the use of *sequence-based neural networks*, e.g., Recurrent Neural Networks, Long Short Term Memory models (Hochreiter and Schmidhuber, 1997), because they can capture long-range temporal dependencies. Tai et al. (2015) introduced Tree-LSTM, a generalisation of LSTMs to *tree-structured network* topologies, e.g., Recursive Neural Networks (Socher et al., 2011). However, this type of network requires input from an external resource (i.e., dependency/constituency parser).

More complex models involved stacked architectures of the three basic forms above (He and Lin, 2016; Yin et al., 2015; Cheng and Kartsaklis, 2015; Zhang et al., 2015; He et al., 2015) which capture the syntactic and semantic structure of a language. However, in addition to being computationally intensive, most of these architectures model sentences as vectors with a fixed size, they risk losing information especially when input sentence vectors are of varying lengths. Recently, Bahdanau et al. (2014) introduced the concept of attention, originally in the context of machine translation, where the network learns to align parts

[2] We use *items* instead of *words* to generalise amongst various representations.

of the source sentence that match the constituents of the target sentence, without having to explicitly form these parts as hard segments. This enables phrase-alignments between sentences as described by Yin and Schütze (2016) in the context of a textual entailment recognition task.

3 Preliminary Experiments

In this section, we describe the preliminary experiments we conducted in order to gain deeper understanding on the limitations of the state-of-the-art model.

Firstly, we define sentence similarity as a supervised learning task where each training example consists of a pair of sentences $(x_1^a, ..., x_{T_a}^a)$, $(x_1^b, ..., x_{T_b}^b)$ of fixed-sized vectors (where $x_i^a, x_j^b \in \mathbb{R}^{d_{input}}$ denoting constituent vectors from each sentence, respectively, which may be of different lengths $T_a \neq T_b$) along with a single real-valued label y for the pair. We evaluated the performance of the state-of-the-art model on this task.

3.1 Model Overview

Since we focus on end-to-end sentence modelling, we implement a simplified (see Table 1) version of MaLSTM (Mueller and Thyagarajan, 2016), i.e., the state-of-the-art model on this task (see Appendix Figure C.1). The model uses a siamese architecture of Long-Short Term Memory (LSTM) to read word vectors representing each input sentence. Each LSTM cell has four components: input gate i_t, forget gate f_t, memory state c_t, and output gate o_t; which decides the information to retain or forget in a sequence of inputs. Equations 1-6 are the updates performed at each LSTM cell for a sequence of input $(x_1, ..., x_T)$ at each timestep $t \in \{1, ..., T\}$, parameterised by weight matrices $W_i, W_f, W_c, W_o, U_i, U_f, U_c, U_o$ and bias vectors b_i, b_f, b_c, b_o.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{1}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{2}$$

$$\tilde{c_t} = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{3}$$

$$c_t = i_t \odot \tilde{c_t} + f_t \odot c_{t-1} \tag{4}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{5}$$

$$h_t = o_t \odot \tanh(c_t) \tag{6}$$

This model computes the sentence similarity based on the Manhattan distance between the final hidden state representations for each sentence: $g(h_{T_a}^a, h_{T_b}^b) = exp(-||h_{T_a}^a - h_{T_b}^b||_1) \in [0, 1]$, which was found to perform better empirically than other simple similarity functions such as cosine similarity (Mueller and Thyagarajan, 2016).

Model Feature	MaLSTM	Ours
pre-training	yes	no
synonym augmentation	yes	no
prediction calibration	yes	no
optimisation method	Adadelta	Adam

Table 1: Model comparison between MaLSTM and our implementation

3.2 Training Details

We use the 300-dimensional pre-trained *word2vec*[3](Mikolov et al., 2013b) word embeddings and compare the performance with that of *GloVe*[4] (Pennington et al., 2014) embeddings. Out-of-embedding-vocabulary (OOEV) words are replaced with an <*unk*> token. We retain the word cases and keep the digits. For character representation, we *fine-tune* the 50-dimensional initial embeddings, modifying them during gradient updates of the neural network model by back-propagating gradients. The chosen size of the embeddings was found to perform best after initial experiments with different sizes.

Our model uses 50-dimensional hidden representations h_t and memory cells c_t. Optimisation of the parameters is done using the SGD-based Adam method (Kingma and Ba, 2014) and we perform gradient clipping to prevent exploding gradients. We tune the hyper-parameters on the validation set by random search since it is infeasible to do a random search across the full hyper-parameter space due to time constraints. After conducting initial experiments, we found the optimal training parameters to be the following: batch size = 30, learning rate = 0.01, learning rate decay = 0.98, dropout = 0.5, number of LSTM layers = 1, maximum epochs = 10, patience = 5 epochs. Patience is the early stopping condition based on performance on validation sets. We used the Tensorflow[5] library to implement and train the model.

3.3 Dataset and Evaluation

We measure the model's performance on three benchmark datasets, i.e., SICK 2014 (Marelli et al., 2014), STS 2016 (Agirre et al., 2016a) and

[3] code.google.com/p/word2vec

[4] https://nlp.stanford.edu/projects/glove/

[5] https://www.tensorflow.org/

Dataset	Baseline	LSTM			Vector Sum		
		GloVe	word2vec	char	GloVe	word2vec	char
SICK 2014	0.5675	0.7430	0.7355	0.3487	0.4903	0.5099	0.0178
PIT 2015	0.4001	0.1187	0.0581	0.0086	0.1263	0.0845	0.0000
STS 2016	0.4757	0.3768	0.2592	0.1067	0.5052	0.4865	-0.0100

Table 2: Pearson Correlation. Performance comparison across input representations and composition models. Baseline method uses cosine similarity measure to predict similarity between sentences.

Dataset	Vocab Size	% OOEV	
		word2vec	GloVe
SICK 2014	2,423	1.4	1.1
PIT 2015	15,156	16.5	9.6
STS 2016	18,061	11.1	7.3

Table 3: Percentage of Out-of-Embedding-Vocabulary (OOEV) words

PIT 2015 (Xu et al., 2015), using Pearson correlation. We assert that a robust model should perform consistently well in these three datasets.

Furthermore, using the framework described in Section 2.1, we chose to compare the model performance at two levels of input representation (i.e., character-level vs word-level) and composition models (i.e., LSTM vs vector sum) in order to eliminate the need for external tools such as parsers.

4 Results and Discussion

Table 2 shows the performance across input representations and composition models. As expected, our simplified model performs relatively worse (Pearson correlation = 0.7355) when compared to what was reported in the original MaLSTM paper (Pearson correlation = 0.8822) on the SICK dataset (using *word2vec*). This performance difference (around 15%) could be attributed to the additional features (see Table 1) that the state-of-the-art model added to their system.

With respect to input representation, the word-based one yields better performance in all datasets over character-level representation for the obvious reason that it carries more semantic information. Furthermore, the character-level representation using LSTM performs better than using *Vector Sum (VS)* because it is able to retain sequential information. Regarding word embeddings, *GloVe* resulted in higher performance compared to *word2vec* in all datasets and models except with VS on the SICK dataset where *word2vec* is slightly better. Table 3 shows the percentage of OOEV words in each dataset with respect to its vocabulary size. Upon closer inspection, we found out that *word2vec* does not have embeddings for stopwords (e.g., *a, to, of, and*). With respect to token-based statistics, these OOEVs comprised 95% (SICK), 67% (PIT) and 44% (STS) respectively in each dataset. Although further work is needed to ascertain the effect of this type of OOEVs, we hypothesise that GloVe's superior performance could be attributed to it, if not to its word vector quality as claimed by Pennington et al. (2014).

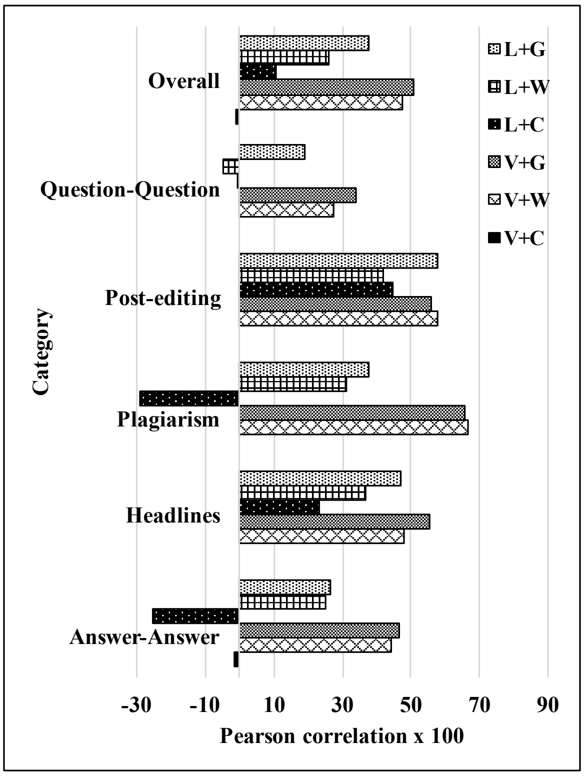

Figure 1: Pearson correlation in STS 2016 evaluation sets (*Key: L=LSTM, V=Vector Sum, C=char, W=word2vec, G=GloVe*)

Dataset	Sentence Pair	Gold Label	Vector Sum	MaLSTM
SICK	A person is scrubbing a zucchini The woman is cutting cooked octopus	0.046	0.523	0.694
STS Question-Question	What kind of socket is this ? What kind of bug is this ?	0.000	0.668	0.778
STS Answer-Answer	You should do it You should never do it	0.200	0.818	0.900
PIT	That s what were watching in Europe If only England was in Europe	0.000	0.566	0.519

Table 4: Examples of difficult sentence pairs. Compositional models use GloVe embeddings.

With respect to the composition model, LSTM performs better than VS but only in the SICK dataset while VS dominates in both the PIT and STS datasets. Specifically, Figure 1 shows the overall and the per-category performance of the model on the STS dataset. Overall, we can clearly see that VS outperforms LSTM by a considerable margin and also in each category except in *Post-editing* and *Headlines*. On the one hand, this suggests that simple compositional models can perform competitively on clean and noisy datasets (e.g., less OOEVs). On the other hand, this shows the ability of LSTM models to capture long term dependencies especially on clean datasets (e.g., SICK dataset) because they contain sufficient semantic information while their performance decreases dramatically on noisy data or on datasets with high proportion of OOEVs (e.g., PIT and STS datasets).

The worst performance was obtained on the PIT dataset in both the baseline[6] and composition models. Aside from PIT dataset's comparatively higher percentage of OOEV words (see Table 3), its diverse, short and noisy user-generated text (Strauss et al., 2016)—typical of social media text—make it a very challenging dataset.

To better understand the reason behind the performance drop of the model, we extracted the 100 most difficult sentence pairs in each dataset by ranking all of the pairs in the test set according to the absolute difference between the gold standard and predicted similarity scores.

We observed that around 60% of the difficult sentence pairs share many similar words (except for a word or two) or contain OOEV words that led to a complete change in meaning. Meanwhile the rest are sentence pairs which are topically similar but completely mean different.

In Table 4, we show examples from each dataset and their corresponding scores (i.e., Pearson correlation) from the gold standard and the composition models. The two sentences come from an actual pair in the dataset.

Example 1 (from SICK dataset) shows a pair of sentences which, although can be interpreted to come from the same domain *food preparation*, are semantically different in their *verb*, *subject*, and *direct object*, for which, presumably, they were labelled in the gold standard as highly dissimilar. However, both of the word-based models predicted them to be highly similar (in varying degrees). This limitation can be attributed to the relatedness of their words (e.g., *person* vs *woman*, *cutting* vs *scrubbing*). Under the *distributional hypothesis assumption* (Harris, 1940; Firth, 1957), two words will have high similarity if they occur in similar contexts even if they neither have the same nor similar meanings. Since word embeddings are typically generated based on this assumption, the relatedness aspect is captured more than genuine similarity. Furthermore, the higher similarity obtained by the LSTM model over Vector Sum can be attributed to its ability to capture syntactic structure in sequences such as sentences.

Examples 2 and 3 (from STS dataset) show sentence pairs which were labelled as completely dissimilar but were predicted with high similarity in both models. This shows the inability of the models to put more weight on semantically rich words which change the overall meaning of a sentence when compared with another.

Example 4 (from PIT dataset) shows a sentence pair which was labelled as completely dissimi-

[6] We represent each sentence with term-frequency vectors.

lar, presumably because it lacks sufficient context for meaningful interpretation. However, they were predicted to some degree as similar possibly because some words are common to both sentences and some are likely related by virtue of co-occurrence in the same context (e.g., *England, Europe)*. See Appendix B for more examples.

5 Future Work

This work is intended to serve as an initial study on end-to-end sentence modelling to identify the limitations associated with it. The models and representations compared, while typical of current sentence modelling methods, are not an exhaustive set and some variations exist. A natural extension to this study is to explore other input granularity representations and composition models presented in the framework. For example, in this study we did not go beyond word representations; however, multi-word expressions are common occurrences in the English language. This could be addressed by modelling sentence constituents using recursive tree structures (Tai et al., 2015) or by learning phrase representations (Wieting et al., 2015).

The limitations of the current word embeddings as revealed in this paper has been studied in the context of word similarity tasks (Levy and Goldberg, 2014; Hill et al., 2016) but to our knowledge had never been investigated explicitly in the context of sentence similarity tasks. For example, Kiela et al. (2015) have shown that specialising semantic spaces to downstream tasks and applications requiring similarity or relatedness can improve performance. Furthermore, some studies (Faruqui et al., 2014; Yu and Dredze, 2014; Ono et al., 2015; Ettinger et al., 2016) have proposed to learn word embeddings by going beyond the distributional hypothesis assumption either through a retrofitting or joint-learning process with some using semantic resources such as ontologies and entity relation databases. Thus, we will explore this direction as this will be particularly important in semantic processing since entities encode much of the semantic information in a language.

Furthermore, the inability of the state-of-the-art model to encode semantically rich words (e.g., *socket*, *bug* in Example 2) with higher weights relative to other words, supports the assertion of Blacoe and Lapata (2012) that distributive semantic representation and composition must be mutually learned. Wieting et al. (2015) have showed that this kind of weighting for semantic importance can be learned automatically when training on a paraphrase database. Recent models (Hashimoto et al., 2016) proposed end-to-end joint modelling at different linguistic levels of a sentence (i.e. morphology, syntax, semantics) on a hierarchy of tasks (i.e., POS tagging, dependency parsing, semantic role labelling)—often done separately—with the assumption that higher-level tasks benefit from lower-level ones.

References

Naveed Afzal, Yanshan Wang, and Hongfang Liu. 2016. Mayonlp at semeval-2016 task 1: Semantic textual similarity based on lexical semantic net and deep learning semantic model. *Proceedings of SemEval* pages 674–679.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, and Janyce Wiebe. 2016a. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation*. pages 509–523.

Eneko Agirre, Aitor Gonzalez-Agirre, Inigo Lopez-Gazpio, Montse Maritxalar, German Rigau, and Larraitz Uria. 2016b. Semeval-2016 task 2: Interpretable semantic textual similarity. *Proceedings of SemEval* pages 512–524.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

Rajendra Banjade, Nabin Maharjan, Nobal B Niraula, and Vasile Rus. 2016. Dtsim at semeval-2016 task 2: Interpreting similarity of texts based on automated chunking, chunk alignment and semantic relation prediction. *Proceedings of SemEval* pages 809–813.

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

Dario Bertero and Pascale Fung. 2015. Hltc-hkust: A neural network paraphrase classifier using translation metrics, semantic roles and lexical similarity features. *Proceedings of SemEval* .

Ergun Biçici. 2015. Rtm-dcu: Predicting semantic similarity with referential translation machines. *SemEval-2015* page 56.

William Blacoe and Mirella Lapata. 2012. A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language*

Processing and Computational Natural Language Learning. Association for Computational Linguistics, pages 546–556.

Tomáš Brychcın and Lukáš Svoboda. 2016. Uwb at semeval-2016 task 1: Semantic textual similarity using lexical, syntactic, and semantic information. *Proceedings of SemEval* pages 588–594.

Jianpeng Cheng and Dimitri Kartsaklis. 2015. Syntax-aware multi-sense word embeddings for deep compositional models of meaning. *arXiv preprint arXiv:1508.02354* http://arxiv.org/abs/1508.02354.

Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics* 4:357–370. https://transacl.org/ojs/index.php/tacl/article/view/792.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, page 350.

Cícero Nogueira dos Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *ICML*. pages 1818–1826.

Allyson Ettinger, Philip Resnik, and Marine Carpuat. 2016. Retrofitting sense-specific word vectors using parallel text. In *Proceedings of NAACL-HLT*. pages 1378–1383.

Asli Eyecioglu and Bill Keller. 2015. ASOBEK: Twitter paraphrase identification with simple overlap features and SVMs. *Proceedings of SemEval* .

Manaal Faruqui, Jesse Dodge, Sujay K Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. 2014. Retrofitting word vectors to semantic lexicons. *arXiv preprint arXiv:1411.4166* .

John Rupert Firth. 1957. *Papers in linguistics, 1934-1951*. Oxford University Press.

Gottlob Frege. 1892. Ubersicht und bedeutung. *Journal of Philosophy and Philosophical Criticism* 100:25–50.

Z. S. Harris. 1940. Review of Louis H. Gray, Foundations of Language (New York: Macmillan, 1939). *Language* 16(3):216–231.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2016. A Joint Many-Task Model: Growing a Neural Network for Multiple NLP Tasks. *arXiv preprint arXiv:1611.01587* https://arxiv.org/abs/1611.01587.

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.

Hua He and Jimmy Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of NAACL-HLT*. pages 937–948.

Hua He, John Wieting, Kevin Gimpel, Jinfeng Rao, and Jimmy Lin. 2016. Umd-ttic-uw at semeval-2016 task 1: Attention-based multi-perspective convolutional neural networks for textual similarity measurement .

Felix Hill, Roi Reichart, and Anna Korhonen. 2016. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics* .

Geoffrey E Hinton. 1984. Distributed representations .

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Yangfeng Ji and Jacob Eisenstein. 2013. Discriminative Improvements to Distributional Sentence Similarity. In *EMNLP*. pages 891–896. http://jiyfeng.github.io/papers/ji-emnlp-2013.pdf.

Douwe Kiela, Felix Hill, and Stephen Clark. 2015. Specializing word embeddings for similarity or relatedness. In *EMNLP*. pages 2044–2048.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*. pages 3294–3302.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *ACL (2)*. pages 302–308.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics* 3:211–225.

Simone Magnolini, Anna Feltracco, and Bernardo Magnini. 2016. Fbk-hlt-nlp at semeval-2016 task 2: A multitask, deep learning approach for interpretable semantic textual similarity. *Proceedings of SemEval* pages 783–789.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *LREC*. pages 216–223.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *ACL*. pages 236–244.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *AAAI*. pages 2786–2792.

Masataka Ono, Makoto Miwa, and Yutaka Sasaki. 2015. Word embedding-based antonym detection using thesauri and distributional information. In *HLT-NAACL*. pages 984–989.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. volume 14, pages 1532–43.

Piotr Przybyła, Nhung TH Nguyen, Matthew Shardlow, Georgios Kontonatsios, and Sophia Ananiadou. 2016. Nactem at semeval-2016 task 1: Inferring sentence-level semantic similarity from an ensemble of complementary lexical and sentence-level features. *Proceedings of SemEval* pages 614–620.

Barbara Rychalska, Katarzyna Pakulska, Krystyna Chodorowska, Wojciech Walczak, and Piotr Andruszkiewicz. 2016. Samsung poland nlp team at semeval-2016 task 1: Necessity for diversity; combining recursive autoencoders, wordnet and ensemble methods to measure semantic similarity. *Proceedings of SemEval* pages 602–608.

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 *Advances in Neural Information Processing Systems*. pages 801–809.

Benjamin Strauss, Bethany E Toma, Alan Ritter, Marie-Catherine de Marneffe, and Wei Xu. 2016. Results of the wnut16 named entity recognition shared task. In *Proceedings of the 2nd Workshop on Noisy User-generated Text*. pages 138–144.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. pages 3104–3112.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075* .

Peter D Turney, Patrick Pantel, et al. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research* 37(1):141–188.

Soroush Vosoughi, Prashanth Vijayaraghavan, and Deb Roy. 2016. Tweet2vec: Learning Tweet Embeddings Using Character-level CNN-LSTM Encoder-Decoder. ACM Press, pages 1041–1044. https://doi.org/10.1145/2911451.2914762.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198* .

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Charagram: Embedding words and sentences via character n-grams. *arXiv preprint arXiv:1607.02789* .

Wei Xu, Chris Callison-Burch, and William B Dolan. 2015. Semeval-2015 task 1: Paraphrase and semantic similarity in twitter (pit). *Proceedings of SemEval* .

Wenpeng Yin and Hinrich Schutze. 2016. Discriminative Phrase Embedding for Paraphrase Identification. *arXiv preprint arXiv:1604.00503* http://arxiv.org/abs/1604.00503.

Wenpeng Yin and Hinrich Schütze. 2016. Why and how to pay different attention to phrase alignments of different intensities. *arXiv preprint arXiv:1604.06896* .

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2015. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *arXiv preprint arXiv:1512.05193* .

Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In *ACL (2)*. pages 545–550.

Guido Zarrella, John Henderson, Elizabeth M. Merkhofer, and Laura Strickhart. 2015. MITRE: Seven systems for semantic similarity in tweets. *Proceedings of SemEval* .

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*. pages 649–657.

Jiang Zhao, Tian Tian Zhu, and Man Lan. 2014. Ecnu: One stone two birds: Ensemble of heterogenous measures for semantic relatedness and textual entailment. *Proceedings of the SemEval* pages 271–277.

Yao Zhou, Cong Liu, and Yan Pan. 2016. Modelling Sentence Pairs with Tree-structured Attentive Encoder. *arXiv preprint arXiv:1610.02806* https://arxiv.org/abs/1610.02806.