Character-based Neural Networks for Sentence Pair Modeling

Wuwei Lan and **Wei Xu**
Department of Computer Science and Engineering
Ohio State University
{lan.105, xu.1265}@osu.edu

Abstract

Sentence pair modeling is critical for many NLP tasks, such as paraphrase identification, semantic textual similarity, and natural language inference. Most state-of-the-art neural models for these tasks rely on pretrained word embedding and compose sentence-level semantics in varied ways; however, few works have attempted to verify whether we really need pretrained embeddings in these tasks. In this paper, we study how effective subword-level (character and character n-gram) representations are in sentence pair modeling. Though it is well-known that subword models are effective in tasks with single sentence input, including language modeling and machine translation, they have not been systematically studied in sentence pair modeling tasks where the semantic and string similarities between texts matter. Our experiments show that subword models without any pretrained word embedding can achieve new state-of-the-art results on two social media datasets and competitive results on news data for paraphrase identification.

1 Introduction

Recently, there have been various neural network models proposed for sentence pair modeling tasks, including semantic similarity (Agirre et al., 2015), paraphrase identification (Dolan et al., 2004; Xu et al., 2015), natural language inference (Bowman et al., 2015), etc. Most, if not all, of these state-of-the-art neural models (Yin et al., 2016; Parikh et al., 2016; He and Lin, 2016; Tomar et al., 2017; Shen et al., 2017) have achieved the best performances for these tasks by using pretrained word embeddings, but results without pretraining are less frequently reported or noted. In fact, we will show that, even with fixed randomized word vectors, the pairwise word interaction model (He and Lin, 2016) based on contextual word vector similarities can still achieve strong performance by capturing identical words and similar surface context features. Moreover, pretrained word embeddings generally have poor coverage in social media domain where out-of-vocabulary rate often reaches over 20% (Baldwin et al., 2013).

We investigated the effectiveness of subword units, such as characters and character n-grams, in place of words for vector representations in sentence pair modeling. Though it is well-known that subword representations are effective to model out-of-vocabulary words in many NLP tasks with a single sentence input, such as machine translation (Luong et al., 2015; Costa-jussà and Fonollosa, 2016), language modeling (Ling et al., 2015; Vania and Lopez, 2017), and sequence labeling (dos Santos and Guimarães, 2015; Plank et al., 2016), they are not systematically studied in the tasks that concern pairs of sentences. Unlike in modeling individual sentences, subword representations have impacts not only on the out-of-vocabulary words but also more directly on the relation between two sentences, which is calculated based on vector similarities in many sentence pair modeling approaches (more details in Section 2.1). For example, while subwords may capture useful string similarities between a pair of sentences (e.g. spelling or morphological variations: *sister* and *sista*, *teach* and *teaches*), they could introduce errors (e.g. similarly spelled words with completely different meanings: *ware* and *war*).

To better understand the role of subword embedding in sentence pair modeling, we performed experimental comparisons that vary (1) the type of subword unit, (2) the composition function, and (3) the datasets of different characteristics. We also presented experiments with language modeling as an auxiliary multi-task learning objective, showing consistent improvements. Taken together, subword and language modeling establish

Proceedings of NAACL-HLT 2018, pages 157–163
New Orleans, Louisiana, June 1 - 6, 2018.

new state-of-the-art results in two social media datasets and competitive results in a news dataset for paraphrase identification without using any pretrained word embeddings.

2 Sentence Pair Modeling with Subwords

The current neural networks for sentence pair modeling (Yin et al., 2016; Parikh et al., 2016; He and Lin, 2016; Liu et al., 2016; Tomar et al., 2017; Wang et al., 2017; Shen et al., 2017, etc) follow a more or less similar design with three main components: (a) contextualized word vectors generated via Bi-LSTM, CNN, or attention, as inputs; (b) soft or hard word alignment and interactions across sentences; (c) and the output classification layer. Different models vary in implementation details, and most importantly, to capture the same essential intuition in the word alignment (also encoded with contextual information) – the semantic relation between two sentences depends largely on the relations of aligned chunks (Agirre et al., 2016). In this paper, we used pairwise word interaction model (He and Lin, 2016) as a representative example and staring point, which reported robust performance across multiple sentence pair modeling tasks and the best results by neural models on social media data (Lan et al., 2017).

2.1 Pairwise Word Interaction (PWI) Model

Let $\boldsymbol{w}^a = (\boldsymbol{w}^a_1, ..., \boldsymbol{w}^a_m)$ and $\boldsymbol{w}^b = (\boldsymbol{w}^a_1, ..., \boldsymbol{w}^b_n)$ be the input sentence pair consisting of m and n tokens, respectively. Each word vector $\boldsymbol{w}_i \in \mathbb{R}^d$ is initialized with pretrained d-dimensional word embedding (Pennington et al., 2014; Wieting et al., 2015, 2016), then encoded with word context and sequence order through bidirectional LSTMs:

$$\overrightarrow{\boldsymbol{h}}_i = LSTM^f(\boldsymbol{w}_i, \overrightarrow{\boldsymbol{h}}_{i-1}) \quad (1)$$

$$\overleftarrow{\boldsymbol{h}}_i = LSTM^b(\boldsymbol{w}_i, \overleftarrow{\boldsymbol{h}}_{i+1}) \quad (2)$$

$$\overleftrightarrow{\boldsymbol{h}}_i = [\overrightarrow{\boldsymbol{h}}_i, \overleftarrow{\boldsymbol{h}}_i] \quad (3)$$

$$\boldsymbol{h}^+_i = \overrightarrow{\boldsymbol{h}}_i + \overleftarrow{\boldsymbol{h}}_i \quad (4)$$

where $\overrightarrow{\boldsymbol{h}}_i$ represents forward hidden state, $\overleftarrow{\boldsymbol{h}}_i$ represents backword hidden state, and $\overleftrightarrow{\boldsymbol{h}}_i$ and $\boldsymbol{h}^+_i$ are the concatenation and summation of two directional hidden states.

For all word pairs $(\boldsymbol{w}^a_i, \boldsymbol{w}^b_j)$ across sentences, the model directly calculates word pair interactions using cosine similarity, Euclidean distance, and dot product over the outputs of the encoding layer:

$$D(\overrightarrow{\boldsymbol{h}}_i, \overrightarrow{\boldsymbol{h}}_j) = [cos(\overrightarrow{\boldsymbol{h}}_i, \overrightarrow{\boldsymbol{h}}_j), \quad (5)$$
$$L2Euclid(\overrightarrow{\boldsymbol{h}}_i, \overrightarrow{\boldsymbol{h}}_j),$$
$$DotProduct(\overrightarrow{\boldsymbol{h}}_i, \overrightarrow{\boldsymbol{h}}_j)].$$

The above equation can also apply to other states $\overleftarrow{\boldsymbol{h}}$, $\overleftrightarrow{\boldsymbol{h}}$ and $\boldsymbol{h}^+$, resulting in a tensor $\mathbf{D}^{13\times m\times n}$ after padding one extra bias term. A "hard" attention is applied to the interaction tensor to further enforce the word alignment, by sorting the interaction values and selecting top ranked word pairs. A 19-layer-deep CNN is followed to aggregate the word interaction features and the softmax layer to predicate classification probabilities.

2.2 Embedding Subwords in PWI Model

Our subword models only involve modification of the input representation layer in the pairwise interaciton model. Let $c_1, ..., c_k$ be the subword (character unigram, bigram and trigram) sequence of a word w. The subword embedding matrix is $\mathbf{C} \in \mathbb{R}^{d'*k}$, where each subword is encoded into the d'-dimension vector. The same subwords will share the same embeddings. We considered two different composition functions to assemble subword embeddings into word embedding:

Char C2W (Ling et al., 2015) applies Bi-LSTM to subword sequence $c_1, ..., c_k$, then the last hidden state $\overrightarrow{\boldsymbol{h}}^{char}_k$ in forward direction and the first hidden state $\overleftarrow{\boldsymbol{h}}^{char}_0$ of the backward direction are linearly combined into word-level embedding $\boldsymbol{w}$:

$$\boldsymbol{w} = \boldsymbol{W}_f \cdot \overrightarrow{\boldsymbol{h}}^{char}_k + \boldsymbol{W}_b \cdot \overleftarrow{\boldsymbol{h}}^{char}_0 + \boldsymbol{b} \quad (6)$$

where $\boldsymbol{W}_f$, $\boldsymbol{W}_b$ and $\boldsymbol{b}$ are parameters.

Char CNN (Kim et al., 2016) applies a convolution operation between subword sequence matrix $\mathbf{C}$ and a filter $\mathbf{F} \in \mathbb{R}^{d'\times l}$ of width l to obtain a feature map $\mathbf{f} \in \mathbb{R}^{k-l+1}$:

$$\mathbf{f}_j = tanh(\langle \mathbf{C}[*, j : j + l - 1], \mathbf{F}\rangle + b) \quad (7)$$

where $\langle A, B\rangle = Tr(\boldsymbol{A}\boldsymbol{B}^T)$ is the Frobenius inner product, b is a bias and $\mathbf{f}_j$ is the jth element of $\mathbf{f}$. We then take the max-over-time operation to select the most important element:

$$y_{\mathbf{f}} = \max_j \mathbf{f}_j. \quad (8)$$

Dataset	Training Size	Test Size	# INV	# OOV	OOV Ratio	Source
PIT-2015	11530	838	7771	1238	13.7%	Twitter trends
Twitter-URL	42200	9324	24905	11440	31.5%	Twitter/news
MSRP	4076	1725	16226	1614	9.0%	news

Table 1: Statistics of three benchmark datasets for paraphrase identification. The training and testing sizes are in numbers of sentence pairs. The number of unique in-vocabulary (INV) and out-of-vocabulary (OOV) words are calculated based on the publicly available GloVe embeddings (details in Section 3.2).

After applying q filters with varied lengths, we can get the array $\boldsymbol{w} = [y_1, ..., y_q]$, which is followed by a one-layer highway network to generate final word embedding.

2.3 Auxiliary Language Modeling (LM)

We adapted a multi-task structure, originally proposed by (Rei, 2017) for sequential tagging, to further improve the subword representations in sentence pair modeling. In addition to training the model for sentence pair tasks, we used a secondary language modeling objective that predicts the next word and previous word using softmax over the hidden states of Bi-LSTM as follows:

$$\overrightarrow{E}_{LM} = -\sum_{t=1}^{T-1}(\log(P(w_{t+1}|\overrightarrow{\boldsymbol{m}_t})) \quad (9)$$

$$\overleftarrow{E}_{LM} = -\sum_{t=2}^{T}(\log(P(w_{t-1}|\overleftarrow{\boldsymbol{m}_t})) \quad (10)$$

where $\overrightarrow{\boldsymbol{m}_t} = tanh(\overrightarrow{\boldsymbol{W}}_{hm}\overrightarrow{\boldsymbol{h}}_t)$ and $\overleftarrow{\boldsymbol{m}_t} = tanh(\overleftarrow{\boldsymbol{W}}_{hm}\overleftarrow{\boldsymbol{h}}_t)$. The Bi-LSTM here is separate from the one in PWI model. The language modeling objective can be combined into sentence pair modeling through a joint objective function:

$$E_{joint} = E + \gamma(\overrightarrow{E}_{LM} + \overleftarrow{E}_{LM}), \quad (11)$$

which balances subword-based sentence pair modeling objective E and language modeling with a weighting coefficient γ.

3 Experiments

3.1 Datasets

We performed experiments on three benchmark datasets for paraphrase identification; each contained pairs of naturally occurring sentences manually labeled as paraphrases and non-paraphrases for binary classification: **Twitter URL** (Lan et al., 2017) was collected from tweets sharing the same URL with major news outlets such as @CNN. This dataset keeps a balance between formal and informal language. **PIT-2015** (Xu et al., 2014, 2015) comes from the Task 1 of Semeval 2015 and was collected from tweets under the same trending topic, which contains varied topics and language styles. **MSRP** (Dolan and Brockett, 2005) was derived from clustered news articles reporting the same event in formal language. Table 1 shows vital statistics for all three datasets.

3.2 Settings

To compare models fairly without implementation variations, we reimplemented all models into a single PyTorch framework.[1] We followed the setups in (He and Lin, 2016) and (Lan et al., 2017) for the pairwise word interaction model, and used the 200-dimensional GloVe word vectors (Pennington et al., 2014), trained on 27 billion words from Twitter (vocabulary size of 1.2 milion words) for social media datasets, and 300-dimensional GloVe vectors, trained on 840 billion words (vocabulary size of 2.2 milion words) from Common Crawl for the MSRP dataset. For cases without pretraining, the word/subword vectors were initialized with random samples drawn uniformly from the range [0.05, 0.05]. We used the same hyperparameters in the C2W (Ling et al., 2015) and CNN-based (Kim et al., 2016) compositions for subword models, except that the composed word embeddings were set to 200- or 300- dimensions as the pretrained word embeddings to make experiment results more comparable. For each experiment, we reported results with 20 epochs.

3.3 Results

Table 2 shows the experiment results on three datasets. We reported maximum F1 scores of any point on the precision-recall curve (Lipton et al., 2014) following previous work.

Word Models The word-level pairwise interaction models, even without pretraining (randomzied) or fine-tuning (fixed), showed strong performance across all three datasets. This reflects

[1]The code and data can be obtained from the first and second author's websites.

	Model Variations	pre-train	#parameters	Twitter URL	PIT-2015	MSRP
Word Models	Logistic Regression	–	–	0.683	0.645	0.829
	(Lan et al., 2017)	Yes	9.5M	0.749	0.667	0.834
	pretrained, fixed	Yes	2.2M	0.753	0.632	0.834
	pretrained, updated	Yes	9.5M	0.756	0.656	0.832
	randomized, fixed	–	2.2M	0.728	0.456	0.821
	randomized, updated	–	9.5M	0.735	0.625	0.834
Subword Models	C2W, unigram	–	2.6M	0.742	0.534	0.816
	C2W, bigram	–	2.7M	0.742	0.563	0.825
	C2W, trigram	–	3.1M	0.729	0.576	0.824
	CNN, unigram	–	6.5M	0.756	0.589	0.820
	CNN, bigram	–	6.5M	0.760	0.646	0.814
	CNN, trigram	–	6.7M	0.753	0.667	0.818
Subword+LM	LM, C2W, unigram	–	3.5M	0.760	**0.691**	0.831
	LM, C2W, bigram	–	3.6M	**0.768**	0.651	0.830
	LM, C2W, trigram	–	4.0M	0.765	0.659	0.831
	LM, CNN, unigram	–	7.4M	0.754	0.665	**0.840**
	LM, CNN, bigram	–	7.4M	0.761	0.667	0.835
	LM, CNN, trigram	–	7.6M	0.759	0.667	0.831

Table 2: Results in F1 scores on Twitter-URL, PIT-2015 and MSRP datasets. The best performance figure in each dataset is denoted in **bold** typeface and the second best is denoted by an underline. Without using any pretrained word embeddings, the Subword+LM models achieve better or competitive performance compared to word models.

the effective design of the BiLSTM and word interaction layers, as well as the unique character of sentence pair modeling, where n-gram overlapping positively signifies the extent of semantic similarity. As a reference, a logistic regression baseline with simple n-gram (also in stemmed form) overlapping features can also achieve good performance on PIT-2015 and MSRP datasets. With that being said, pretraining and fine-tuning word vectors are mostly crucial for pushing out the last bit of performance from word-level models.

Subword Models (+LMs) Without using any pretrained word embeddings, subword-based pairwise word interaction models can achieve very competitive results on social media datasets compared with the best word-based models (pretrained, fixed). For MSRP with only 9% of OOV words (Table 1), the subword models do not show advantages. Once the subword models are trained with multi-task language modeling (Subword+LM), the performance on all datasets are further improved, outperforming the best previously reported results by neural models (Lan et al., 2017). A qualitative analysis reveals that subwords are crucial for out-of-vocabulary words while language modeling ensures more semantic and syntactic compatibility (Table 3).

Model	INV Words		OOV Words	
	any	*walking*	*#airport*	*brexit*
Word	*anything*	*walk*	*salomon*	*bollocks*
	anyone	*running*	*363*	*misogynistic*
	other	*dead*	*#trumpdchotel*	*patriarchy*
	there	*around*	*hillarys*	*sexist*
Subword	*analogy*	*waffling*	*@atlairport*	*grexit*
	nay	*slagging*	*#dojreport*	*bret*
	away	*scaling*	*#macbookpro*	*juliet*
	andy	*#hacking*	*#guangzhou*	*#brexit*
Subword + LM	*any1*	*warming*	*airport*	*#brexit*
	many	*wagging*	*#airports*	*brit*
	ang	*waging*	*rapport*	*ofbrexit*
	nanny	*waiting*	*#statecapturereport*	*drought-hit*

Table 3: Nearest neighbors of word vectors under cosine similarity in Twitter-URL dataset.

3.4 Combining Word and Subword Representations

In addition, we experimented with combining the pretrained word embeddings and subword models with various strategies: concatenation, weighted average, adaptive models (Miyamoto and Cho, 2016) and attention models (Rei et al., 2016). The weighted average outperformed all others but only showed slight improvement over word-based models in social media datasets; other combination strategies could even lower the performance. The best performance was 0.763 F1 in Twitter-URL and 0.671 in PIT-2015 with a weighted average: 0.75 $\times$ word embedding + 0.25 $\times$ subword embedding.

4 Model Ablations

In the original PWI model, He and Lin (2016) performed pattern recognition of complex semantic relationships by applying a 19-layer deep convolutional neural network (CNN) on the word pair interaction tensor (Eq. 5). However, the SemEval task on Interpretable Semantic Textual Similarity

	Model Variations	CNN[19]	#parameters	#hours/epoch	Twitter URL	PIT-2015	MSRP
Word Models	Logistic Regression	–	–	–	0.683	0.645	0.829
	pretrained, fixed	Yes	2.2M	4.5h	0.753	0.632	0.834
		–	1.4M	3.2h	0.741	0.602	0.827
Subword Models	C2W, unigram	Yes	2.6M	5.8h	0.742	0.534	0.816
		–	1.4M	4.6h	0.741	0.655	0.808
	CNN, unigram	Yes	6.5M	5.4h	0.756	0.589	0.820
		–	5.3M	4.2h	0.759	0.659	0.809
Subword+LM	LM, C2W, unigram	Yes	3.5M	6.5h	**0.760**	**0.691**	0.831
		–	2.3M	5.3h	0.746	0.625	0.811
	LM, CNN, unigram	Yes	7.4M	5.8h	0.754	0.665	**0.840**
		–	6.2M	4.6h	0.758	0.659	0.809

Table 4: Comparison of F1 scores between the original PWI model with 19-layer CNN for aggregation and the simplified model without 19-layer CNN on Twitter-URL, PIT-2015 and MSRP datasets. The number of parameters and training time per epoch shown are based on the Twitter URL dataset and a single NVIDIA Pascal P100 GPU.

(Agirre et al., 2016) in part demonstrated that the semantic relationship between two sentences depends largely on the relations of aligned words or chunks. Since the interaction tensor in the PWI model already encodes word alignment information in the form of vector similarities, a natural question is whether a 19-layer CNN is necessary.

Table 4 shows the results of our systems with and without the 19-layer CNN for aggregating the pairwise word interactions before the final softmax layer. While in most cases the 19-layer CNN helps to achieve better or comparable performance, it comes at the expense of ∼25% increase of training time. An exception is the character-based PWI without language model, which performs well on the PIT-2015 dataset without the 19-layer CNN and comparably to logistic regression with string overlap features (Eyecioglu and Keller, 2015). A closer look into the datasets reveals that PIT-2015 has a similar level of unigram overlap as the Twitter URL corpus (Table 5),[2] but lower character bigram overlap (indicative of spelling variations) and lower word bigram overlap (indicative of word reordering) between the pairs of sentences that are labeled as paraphrase.

The 19-layer CNN appears to be crucial for the MSRP dataset, which has the smallest training size and is skewed toward very high word overlap.[2] For the two social media datasets, our subword models have improved performance compared to pretrained word models regardless of having or not having the 19-layer CNN.

	Twitter URL	PIT-2015	MSRP
#char unigrams in shorter sentence (all)	67.5	36.2	109.0
#char unigrams in longer sentence (all)	97.7	50.5	128.5
#char unigrams of the union (all)	101.1	53.0	130.0
#char unigrams of the intersection (all)	64.1	33.7	107.4
char unigram overlap (all)	63.4%	63.5%	82.6%
char unigram overlap (paraphrase-only)	68.8%	67.0%	84.7%
char bigram overlap (all)	30.8%	33.6%	67.4%
char bigram overlap (paraphrase-only)	48.2%	42.4%	71.6%
word unigram overlap (all)	13.3%	21.7%	54.8%
word unigram overlap (paraphrase-only)	32.0%	30.2%	59.1%
word bigram overlap (all)	5.3%	8.4%	33.2%
word bigram overlap (paraphrase-only)	17.9%	12.3%	36.8%

Table 5: Character and word overlap comparison.

5 Conclusion

We presented a focused study on the effectiveness of subword models in sentence pair modeling and showed competitive results without using pretrained word embeddings. We also showed that subword models can benefit from multi-task learning with simple language modeling, and established new start-of-the-art results for paraphrase identification on two Twitter datasets, where out-of-vocabulary words and spelling variations are profound. The results shed light on future work on language-independent paraphrase identification and multilingual paraphrase acquisition where pretrained word embeddings on large corpora are not readily available in many languages.

Acknowledgments

We thank John Wieting and Mikhail Belkin for valuable discussions, and Ohio Supercomputer Center (Center, 2012) for computing resources. Funding was provided by the U.S. Army Research Office (ARO) and Defense Advanced Research Projects Agency (DARPA) under Contract Number W911NF-17-C-0095. The content of the information in this document does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation here on.

[2] See more discussions in (Lan et al., 2017).

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval)*.

Eneko Agirre, Aitor Gonzalez-Agirre, Inigo Lopez-Gazpio, Montse Maritxalar, German Rigau, and Larraitz Uria. 2016. SemEval-2016 task 2: Interpretable semantic textual similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval)*.

Timothy Baldwin, Paul Cook, Marco Lui, Andrew MacKinlay, and Li Wang. 2013. How noisy social media text, how diffrnt social media sources? In *Proceedings of the Sixth International Joint Conference on Natural Language Processing (IJCNLP)*.

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 (EMNLP)*.

Ohio Supercomputer Center. 2012. Oakley supercomputer. http://osc.edu/ark:/19495/hpc0cvqn.

Marta R. Costa-jussà and José A. R. Fonollosa. 2016. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*.

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 (COLING)*.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP)*.

Cicero dos Santos and Victor Guimarães. 2015. Boosting named entity recognition with neural character embeddings. In *Proceedings of the Fifth Named Entity Workshop*.

Asli Eyecioglu and Bill Keller. 2015. ASOBEK: Twitter paraphrase identification with simple overlap features and SVMs. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval)*.

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 (NAACL-HLT)*.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI)*.

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 on Natural Language Processing (EMNLP)*.

Wang Ling, Tiago Luís, Luís Marujo, Rámon Fernandez Astudillo, Silvio Amir, Chris Dyer, Alan W Black, and Isabel Trancoso. 2015. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Zachary C. Lipton, Charles Elkan, and Balakrishnan Naryanaswamy. 2014. Optimal thresholding of classifiers to maximize F1 measure. In *Proceedings of the 2014th European Conference on Machine Learning & Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*.

Pengfei Liu, Xipeng Qiu, Yaqian Zhou, Jifan Chen, and Xuanjing Huang. 2016. Modelling interaction of sentence pair with coupled-LSTMs. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*.

Yasumasa Miyamoto and Kyunghyun Cho. 2016. Gated word-character recurrent language model. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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 (EMNLP)*.

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

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Marek Rei. 2017. Semi-supervised multitask learning for sequence labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Marek Rei, Gamal Crichton, and Sampo Pyysalo. 2016. Attending to characters in neural sequence labeling models. In *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers (COLING)*.

Gehui Shen, Yunlun Yang, and Zhi-Hong Deng. 2017. Inter-weighted alignment network for sentence pair modeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Gaurav Singh Tomar, Thyago Duque, Oscar Täckström, Jakob Uszkoreit, and Dipanjan Das. 2017. Neural paraphrase identification of questions with noisy pretraining. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP (SCLeM)*.

Clara Vania and Adam Lopez. 2017. From characters to words to in between: Do we capture morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI)*.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. From paraphrase database to compositional paraphrase model and back. *Transactions of the Association for Computational Linguistics (TACL)* 3:345–358.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*.

Wei Xu, Chris Callison-Burch, and William B. Dolan. 2015. SemEval-2015 task 1: Paraphrase and semantic similarity in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval)*.

Wei Xu, Alan Ritter, Chris Callison-Burch, William B. Dolan, and Yangfeng Ji. 2014. Extracting lexically divergent paraphrases from Twitter. *Transactions of the Association for Computational Linguistics (TACL)* 2(1).

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2016. ABCNN: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics (TACL)* 4:259–272.