Deep Learning of Binary and Gradient Judgements for Semantic Paraphrase

Yuri Bizzoni
University of Gothenburg
yuri.bizzoni@gu.se

Shalom Lappin
University of Gothenburg
shalom.lappin@gu.se

Abstract

We treat paraphrase identification as an ordering task. We construct a corpus of 250 sets of five sentences, with each set containing a reference sentence and four paraphrase candidates, which are annotated on a scale of 1 to 5 for paraphrase proximity. We partition this corpus into 1000 pairs of sentences in which the first is the reference sentence and the second is a paraphrase candidate. We then train a DNN encoder for sentence pair inputs. It consists of parallel CNNs that feed parallel LSTM RNNs, followed by fully connected NNs, and finally a dense merging layer that produces a single output. We test it for both binary and graded predictions. The latter are generated as a by-product of training the former (the binary classifier). It reaches 70% accuracy on the binary classification task. It achieves a Pearson correlation of .59-.61 with the annotated gold standard for the gradient ranking candidate sets.

1 Introduction

Paraphrase identification is an area of research with a long history. Approaches to the task can be divided into supervised methods, such as (Madnani et al., 2012), currently the most commonly used, and unsupervised techniques (Socher et al., 2011).

While many approaches of both types use carefully selected features to determine similarity, such as string edit distance (Dolan et al., 2004)) or longest common subsequence (Fernando and Stevenson, 2008), several recent supervised approaches apply Neural Networks to the task (Filice et al., 2015; He et al., 2015), often linking it to the related issue of semantic similarity (Tai et al., 2015; Yin and Schütze, 2015).

Traditionally, paraphrase detection has been formulated as a binary problem. Corpora employed in this work contain pairs of sentences labeled as *paraphrase* or *non-paraphrase*. The most representative of these corpora, such as the Microsoft Paraphrase Corpus (Dolan et al., 2004), conform to this paradigm.

This approach is different from the one adopted in semantic similarity datasets, where a pair of words or sentences is labeled on a gradient classification system. In some cases, semantic similarity tasks overlap with paraphrase detection, as in Xu et al. (2015) and in Agirre et al. (2016). Xu et al. (2015) is one of the first works that tries to connect paraphrase identification with semantic similarity. They define a task where the system generates both a binary judgment and a gradient score for sentences pairs.

We present a new dataset for paraphrase identification which is built on two main ideas: (i) Paraphrase recognition is a gradient classification task. (ii) Paraphrase recognition is an ordering problem, where *sets of sentences* are ranked by similarity with respect to a reference sentence.

While the first assumption is shared by some of the work we have cited here, our corpus is, to the best of our knowledge, the first one constructed on the basis of the second claim.

We believe that annotating sets of sentences for similarity with respect to a reference sentence can help with both the learning and the testing processes in paraphrase identification.

We use this corpus to test a neural network architecture formed by a combination of Convolutional Neural Networks (CNNs) and Long Short Term Memory Recurrent Neural Networks (LSTM RNNs). We test this model on two classification problems: (i) binary paraphrase classification, and (ii) paraphrase ranking. We show that our system can achieve a significant correlation to human paraphrase judgments on the ranking task as a by-product of supervised binary learning.

2 A New Type of Corpus for Paraphrase Recognition

At this stage our corpus is formed of 250 sets of five sentences. In each set, the first sentence is the reference sentence, while the remaining four sentences are labeled on a 1-5 scale, based on their degree of paraphrase similarity with respect to the reference sentence. This is on analogy with the annotation frame used for SemEval Semantic Similarity tasks (Agirre et al., 2016). Every group of 5 sentences illustrates (possibly different) graduated degrees of paraphrasehood relative to the reference sentence. Broadly, our labels represent the following categories: (1) Two sentences are completely unrelated. (2) Two sentences are semantically related, but they are not paraphrases. (3) Two sentences are weak paraphrases. (4) Two sentences are strong paraphrases. (5) Two sentences are (type) identical.

The following example illustrates these ranking labels.

- Ref. sent: *A woman feeds a cat*
  - A woman kicks a cat. *Score: 2*
  - A person feeds an animal *Score: 3*
  - A woman is feeding a cat. *Score: 4*
  - A woman feeds a cat . *Score: 5*
- Ref. sent: *I have a black hat*
  - Larry teaches plants to grow. *Score: 1*
  - I have a red hat . *Score: 2*
  - My hat is night black ; pitch black. *Score: 3*
  - My hat's color is black. *Score: 4*

While the extremes of our scale (1 and 5) are relatively rare in our corpus, we focus on the intermediate cases of paraphrase, from non-paraphrases with some semantic similarity (2) to non type-identical strong paraphrases (4).

We believe that this annotation scheme is particularly useful. While it sustains graded semantic similarity labels, it also provides sets of semantically related elements, each one of which can be scored or ordered independently from the others. Therefore, the reference sentence can be tested separately for each sentence in the set in a binary classification task. In the test phase, this annotation schema allows us to observe how a system represents the similarity between two sentences by taking the scores of two candidates as points of relative proximity to the reference sentence.

Our examples above indicate that a binary classification can be misleading because it conceals the different levels of similarity between competing candidates.

We find instead that framing paraphrase recognition as an ordering problem allows a more flexible evaluation of a model. It permits us to evaluate the relative proximity of several candidate paraphrases to the reference sentence independently of the particular paraphrase score that the model assigns to each candidate in the set.

For example, the sentence *A person feeds an animal* can be considered to be a loose paraphrase of the sentence **A woman feeds a cat**, or alternatively, as a semantically related non-paraphrase. Which of these conclusions we adopt depends on our decision concerning how much content sentences need to share in order to be classified as paraphrases. By contrast, it would be far fetched to suggest that *A woman kicks a cat* is a better or even equally strong paraphrase for **A woman feeds a cat**. Similarly, the sentences **I have a black hat** and *My hat is night black* can be considered to be loose paraphrases, or semantically related non-paraphrases. But *I have a red hat* cannot plausibly be taken as more similar in meaning to **I have a black hat** than *My hat is night black*.

The core of this dataset was built from various parts of the Brown Corpus (Francis and Kucera, 1979), mainly from the news and narrative sections. For each sentence, we introduced raw paraphrases by round trip machine translation from English through Swedish, German, Spanish and Japanese, back to English. This process yielded paraphrases, looser relations of semantic relatedness, and non-paraphrases.

One of the authors then manually annotated each set of five sentences and corrected grammatical infelicities. We also introduced more interesting syntactic and semantic variation. For example we manually constructed many cases of negation and passive/active mood switch. This allows us to test paraphrase over a wider range of syntactic and lexical semantic constructions. Similar manually generated elements were often substituted as candidate paraphrases to round-trip generated candidates judged to be of little interest for the task. So, for example, we frequently had several strong paraphrases produced by round-trip translation, resulting in groups of three or four strong candidates for a reference sentence, and we replaced several of these with our own alternatives.

A number of shorter examples produced by the authors were also added to the corpus. These are intended to test the performance of the system for specific semantic relations, such as antinomy (*I have a new car – I have an old car*), expansion (*His car is red – His car has a characteristic red colour*) and subject–object permutation (*A white blanket covered her mouth – Her mouth covered a white blanket*).

One of the authors assigned the 1-5 ratings for each sentence in a reference set. We naturally regard this as a ”weak” point in our dataset. As we discuss in the Conclusion, we intend to use crowd sourcing to obtain more broadly based and reliable speaker annotation for our examples.

Our corpus has the advantage of being suitable for both training a binary classifier and developing a model to predict gradient paraphrase judgments. For the former, we simply consider every score over a given gradient threshold label as 1, and scores below that threshold as 0. For gradient classification we use all the scoring labels to test the correlation between a system's ordering performance and our human judgments. We will show how, once a model has been trained for a binary detection task, we can check its performance on the gradient ordering task.

3 A DNN for Paraphrase Classification

For classification and gradient judgment prediction we constructed a deep neural network. Its architecture consists of three main components:

1. Two encoders that learn the representation of two sentences separately
2. A unified layer that merges the output of the encoders
3. A final set of fully connected layers that work on the merged representation of the two sentences to generate a judgment.

The encoder for each pair of sentences taken as input is composed of two parallel Convolutional Neural Networks and LSTM RNNs, feeding two sequenced fully connected layers.

The first layer of our encoders is a CNN with 50 filters of length 5. CNNs have been successfully applied to problems in computational semantics, such as text classification and sentiment analysis (Lai et al., 2015), as well as to paraphrase recognition (Socher et al., 2011). In this part of our model, the encoder learns a more compact representation of the sentence, with reduced vector space dimensions and features. This permits the NN to focus on the information most relevant to paraphrase identification.

We use an ”Atrous” Convolutional Neural Network (Giusti et al., 2013; Chen et al., 2016). An ”Atrous” CNN is a modified form of Convolutional Network designed to reduce the risk of losing important information in max pooling. In the case of a standard CNN, max pooling will perform a reduction of the output of the convolutional layer, selecting only some information contained in it. In the case of image processing, for example, a 2x2 max pooling on the so-called ”map” returned by the convolutional layer will create a smaller map that does not contain information from the entire original map, but only from a specific region of such map, or mirroring a specific pattern in the original image: for example, all the patches whose upper left corner lies on even coordinates on the map (Giusti et al., 2013). This way of processing information can undermine the results when complex inputs are involved. An Atrous network fragments the map returned by the max pooling layer, so that each fragment contains information independent of the other fragments, and each reduced map contains information from all the patches of the input. This is a good strategy for speeding up processing time by avoiding redundant computation.

The output of each CNN is passed through a max pooling layer to an LSTM RNN. Since the CNN and the max pooling layer perform discriminative reduction of the input dimensionality, we can run a large LSTM RNN model (50 smart cells) without substantial computational cost. In this phase of processing, the vector dimensions of the sentence representation is further reduced, with relevant information (hopefully) conserved and highlighted, particularly for the sequential structure of the data. Each encoder is completed by two successive fully connected layers of dimensions 50 and 300, respectively, that produces a vector representation for an input sentence in the pair. The first one has a .5 dropout rate.

The 300 dimensional outputs of the two encoders are then passed to a layer that merges them into a single vector. We found that simple vector concatenation was the best option for performing this merge. To measure the similarity of two sequences our model only makes use of the information contained in the merged version of the encoders' output. We did not use a device in the merging phase to assess similarity between two sequences. The merging layer feeds the concatenated input to a series of five fully connected layers. The last layer applies a sigmoid function to produce the classifier judgment. While the sigmoid function performs well for binary classification, it returns a gradient over its input, thus generating an ordering of values for the ranking task.

These three kinds of Neural Network capture information in different ways. They can be combined to achieve a better global representation of sentence input. Specifically, while a CNN can reduce the spectral variance of input, an LSTM RNN is designed to model its sequential dimension over time. The CNN manages to reduce the input's dimensionality while keeping the ordering information of the original sentence. This information will then be processed by the LSTM RNN, which is particularly well suited for handling words sequenced through time.

Also, an LSTM RNN's performance can be strongly improved by providing it with better features (Pascanu et al., 2014). In our case this is accomplished by the CNN. The densely connected layers create clearer, more separable final vector representations of the data. To encode the original sentences we used Word2Vec embeddings pre-trained on Google News (Mikolov et al., 2013).

Table 1 gives the binary accuracy, and ranked ordering Pearson correlation performance of our model, over 10 fold validation, after 200 epochs.

Table 2 presents accuracy and F1 for different versions of our model. The baseline is the model's performance without any training. We compute the baseline by relying solely on the pre-loaded Word2Vec lexical embedding content of the words' distributional vectors to obtain a semantic similarity judgment. No learning from our corpus annotation is involved. The sentence's vectors are still reduced to a single vector through the LSTM layer, but this is done without corpus based supervision or training.

4 Binary Classification Task

To use our corpus for a binary classification task we map each set of five sentences into a series of pairs, where the first element is the reference sentence and the second element is one of the four remaining sentences. Gradient labels are then replaced by binary ones. We consider all labels higher than 2 as positive judgments (Paraphrase) and all labels equal to or lower than 2 as negative judgments (Non-Paraphrase). We train our model with these labels for a binary classification task.

We split our corpus into a training and a test set, making sure that the two sets contained completely distinct reference-candidate pairs. While a small minority of reference sentences is the same in train and test, their candidate paraphrases are always different.

We ran the training phase for 200 epochs, keeping the order of the input fixed (due to curriculum learning issues). Training on 761 pairs of sentences and testing on 239 pairs, we reached an average

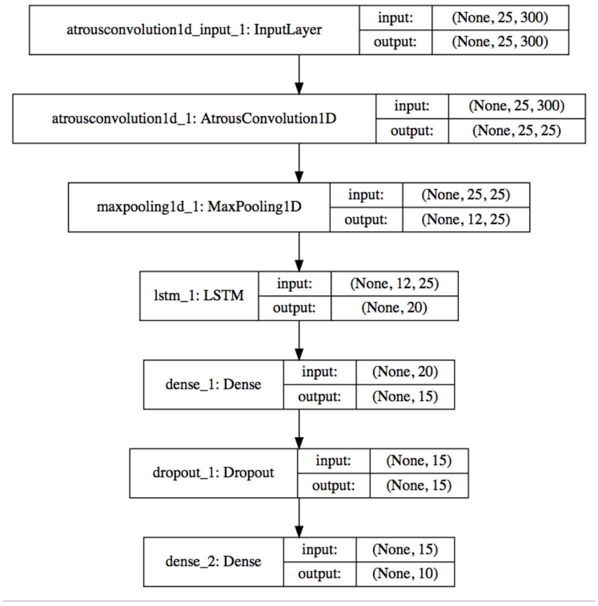

Figure 1: Example of an encoder. A padded input of fixed length is passed to a CNN, a max pooling layer, a single LSTM RNN, and finally two fully connected layers separated by a dropout layer of 0.5. The input's and output's shape is indicated in brackets for each layer

accuracy of 70.0 over 10 fold cross-validation. We see that our architecture learned to recognize different semantic and syntactic phenomena with a promising level of accuracy, although it is not state of the art in paraphrase recognition for systems trained on large corpora, such as the Microsoft Paraphrase Corpus (Ji and Eisenstein, 2013). [1]

A small corpus may cause instability in results. Interestingly, we found that our DNN is able to generalize consistently on the following patterns:

- **Negation**. This is a rich man's world – This is not a rich man's world. *Non-Paraphrase*;
- **Subject–Object permutation**. The man follows the wolf – The wolf follows the man. *Non-Paraphrase*;
- **Active–Passive relation**. A white blanket covered her mouth – Her mouth was covered with a white blanket. *Paraphrase*;
- **Various cases of loose paraphrase** The man follows the wolf – The person follows the animal. *Paraphrase*.

However, our model had trouble with several others cases, some due to its lack of relevant world knowledge, and others because of its limited capacity for semantically driven inference. These include:

- **Time expressions**. It was morning – It was noon. *Non-Paraphrase*;
- **Some cases of antinomy**. This is not good – This is bad. *Paraphrase*;
- **Space expressions**. Some years ago I was going to school when I met a man – Some years ago I was going to church when I met a man. *Non-paraphrase*.

Predictably, the model has difficulty in learning a pattern or a phrase when it is under represented in the training data. In some cases, the effect of data scarcity can be observed in an ”overfit weighting” of specific words. We believe that these idiosyncrasies can be overcome through training on a larger set.

[1] This is to be expected, given the specific nature of the task and the small dimensions of our dataset. It is also worth noting that, while sentences in the Microsoft Paraphrase Corpus are generally longer, our corpus contains a much larger variety of syntactic and semantic patterns, including ”more difficult” cases, like passive-active change and negation.

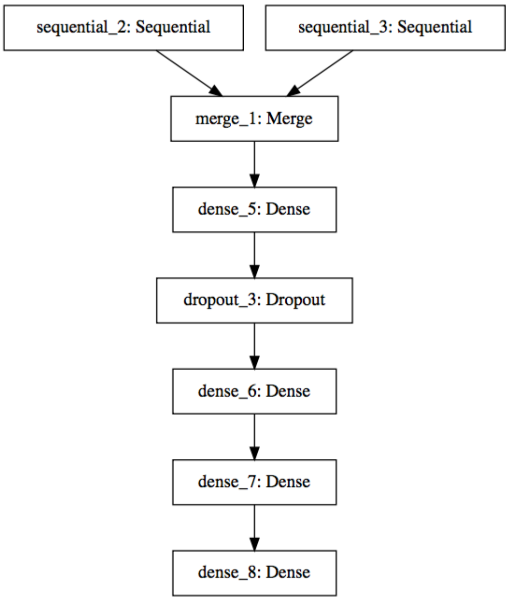

Figure 2: A more abstract representation of our full model. Sequential 2 and sequential 3 are encoders of the kind specified in Figure 1. Their outputs are concatenated in merge 1 and fed to a series of dense layers. Dropout 3 has a rate of 0.2

We observe that, on occasion, the model's errors are in the gray area between clear paraphrase and clear non-paraphrase. Here the correctness of a label is not obvious. For example, the pair *I am so sleepy I can barely stand – I am sleep deprived* can be considered to be a loose paraphrase pair, or they can be taken as an instance of non-paraphrase.

5 Paraphrase Ordering Task

Once the DNN has learned representations for binary classification, we can use it to rank the sentences of the test set in order of paraphrase proximity. We apply the sigmoid value distribution for the candidate sentences in a set of five (the reference and four candidates) to determine the ranking. To do this we use the original structure of our dataset, composed of sets of five sentences.

First, we attribute a similarity score to all pairs of sentences (reference sentence and candidate paraphrase) in a set. This is the similarity score learned in the binary task, which is determined by the sigmoid function applied on the output of our DNN. In this case, we don't "round" the judgment, as we are not seeking a 0,1 output.

We compute the average Pearson correlation on all sets of the test corpus to check the extent to which the ranking that our algorithm produces matches our gradient annotation. We found comparable and meaningful correlations between our ranking and the ordering that our model predicts. These correlations indicate that our model achieves an encouraging level of accuracy in predicting our gradient annotations for the candidate sentences in a set, using the weights learned for a binary classification task.

This task differs from the binary classification task in several important respects. In one way, it is easier. A non-paraphrase can be misjudged as a paraphrase and still fall in the right order within a ranking. In another sense, it is more difficult. Strict paraphrases, loose paraphrases, and semantically similar non-paraphrases have to be ordered in accord with human judgment patterns, which is a more complex task than simple binary classification.

Our gradient ranking system allows us to have a more nuanced view concerning some of the issues that arise in pairwise paraphrase labeling that we pointed out at the end of the previous section.

The existence of a correlation between our annotation ordering and our model's predictions is a by-product of supervised binary learning. Since we are re-using the representations learned for the binary

k	Accuracy
1	70.10
2	67.01
3	79.38
4	73.20
5	67.01
6	72.92
7	66.67
8	75.79
9	64.21
10	73.68

k	Pearson
1	.51
2	.63
3	.59
4	.62
5	.61
6	.72
7	.59
8	.67
9	.54
10	.67

Table 1: Accuracy (*on the binary task*) and Pearson Correlation (*on the ordering task*) Over Ten Fold Validation Testing after 200 epochs. The accuracy reported in the paper is an average over these results.

Model	Accuracy	F1
Baseline (without training)	42.1	59.3
Our model	**78.0**	74.6
Encoders without LSTM	65.9	68.9
Encoders without ACNN	69.5	50.8
Just one layer after concatenation	73.0	70.0
Using CNN instead of ACNN	76.6	76.0
ACNN with 10 filters	70.4	68.1
LSTM with 10 filters	69.0	71.3
Without dropouts	72.6	71.0
Merging via multiplication	72.6	71.1
Encoders without dense layers	72.2	71.7

Table 2: Accuracy for different versions of the model after 200 epochs. Each model ran on our standard train and test data, *without* our performing cross-validation.

task in order to perform a new task, we consider it a form of transfer learning from a supervised binary context (assigning a 0/1 value to a pair of sentences) to an unsupervised ordering problem (ranking a set of sentences). In this case, our corpus allowed us to perform double transfer learning. First, we use word embeddings trained to maximize single words' contextual similarity, in order to train on a supervised binary paraphrase dataset. Then, we use the representations acquired in this way to perform an ordering task for which the DNN has not been trained.

The fact that ranked correlations are sustained through binary paraphrase classification is not an obvious result. A model trained on $\{0,1\}$ labels could ”polarize” its scores to the point where no meaningful ordering would be available. Had this happened, a good performance in a binary task would actually conceal the loss of important semantic information. Xu et al. (2015), discussing the relation of paraphrase identification to the recognition of semantic similarity, observe that there is no necessary connection between binary classification and prediction of gradient labels, and that an increase in one can even produce a loss in the other.

6 Conclusions and Future Work

We present a new kind of corpus to evaluate paraphrase identification and we construct a novel type of DNN architecture for a set of paraphrase classification tasks. We show that our model learns an effective representation of sentences for such paraphrase tasks.

Our corpus' design is based on the assumption that paraphrase ranking is a useful way to approach the paraphrase identification problem. We show how this kind of corpus can be used for supervised learning of binary classification, for multi-class classification, and for gradient judgment prediction.

The neural network architecture that we propose encodes each sentence in a low dimensional representation, combining a CNN, an LSTM RNN, and two densely connected neural layers. The two output representations of the encoders are then merged through concatenation, and fed to a series of densely connected layers.

While binary classification is directly learned in the training phase, our model also yields a robust correlation to human judgments in the ordering task through the softmax sigmoid distributions generated for binary classification. While the model learns to classify two sentences as paraphrases or non-paraphrases, it retains enough information to assign gradient values to members of sets of sentences in a way that correlates significantly with our annotation.

Our model doesn't use any "alignment" of the data. The encoders' representations are simply concatenated. This gives our DNN considerable flexibility in modeling patterns such as subject–object permutation (*The man follows the wolf – The wolf follows the man*), and sentence expansions (*A man eats the food – There is a man and he eats the food*). It can also create complications where a simple alignment of two sentences might suffice to identify a similarity. We will experiment with the addition of some form of alignment to our model in future work.

We will be experimenting with crowd sourcing to obtain more reliable annotation of our corpus. We will also be expanding the corpus to encompass a wider range of syntactic and semantic patterns, and to include a significantly larger number of reference + candidate sets. Finally, we will be looking at alternative DNN architectures, particularly those with attentional components, in an effort to improve the performance of our models for both the binary classification and gradient judgment prediction tasks.

7 Acknowledgments

We are grateful to three anonymous reviewers for helpful comments and suggestions on an earlier draft of this paper. The research reported in this paper was supported by a grant from the Swedish Research Council for the establishment of the Centre for Linguistic Theory and Studies in Probability (CLASP) at the University of Gothenburg. We would also like to thank our colleagues at CLASP at the University of Gothenburg for useful discussion of many of the ideas presented here. We are solely responsible for any errors which may remain in this paper.

References

Agirre, E., C. Banea, D. M. Cer, M. T. Diab, A. Gonzalez-Agirre, R. Mihalcea, G. Rigau, and J. 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@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pp. 497–511.

Chen, L., G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille (2016). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR abs/1606.00915*.

Dolan, B., C. Quirk, and C. Brockett (2004). Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Fernando, S. and M. Stevenson (2008). A semantic similarity approach to paraphrase detection. *Computational Linguistics UK (CLUK 2008) 11th Annual Research Colloqium.*

Filice, S., G. Da San Martino, and A. Moschitti (2015). *Structural representations for learning relations between pairs of texts*, Volume 1, pp. 1003–1013. Association for Computational Linguistics (ACL).

Francis, W. N. and H. Kucera (1979). Brown corpus manual. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US.

Giusti, A., D. C. Ciresan, J. Masci, L. M. Gambardella, and J. Schmidhuber (2013). Fast image scanning with deep max-pooling convolutional neural networks. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pp. 4034–4038. IEEE.

He, H., K. Gimpel, and J. Lin (2015, September). Multi-perspective sentence similarity modeling with convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, pp. 1576–1586. Association for Computational Linguistics.

Ji, Y. and J. Eisenstein (2013). Discriminative improvements to distributional sentence similarity. In *In EMNLP*, pp. 891–896.

Lai, S., L. Xu, K. Liu, and J. Zhao (2015). Recurrent convolutional neural networks for text classification. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pp. 2267–2273. AAAI Press.

Madnani, N., J. Tetreault, and M. Chodorow (2012). Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, Stroudsburg, PA, USA, pp. 182–190. Association for Computational Linguistics.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean (2013). Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 26*, pp. 3111–3119. Curran Associates, Inc.

Pascanu, R., C. Gulcehre, K. Cho, and Y. Bengio (2014). *How to construct deep recurrent neural networks*.

Socher, R., E. H. Huang, J. Pennington, A. Y. Ng, and C. D. Manning+ (2011). Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection. In *Advances in Neural Information Processing Systems 24*.

Tai, K. S., R. Socher, and C. D. Manning (2015). Improved semantic representations from tree-structured long short-term memory networks. *CoRR abs/1503.00075*.

Xu, W., C. Callison-Burch, and B. Dolan (2015, June). Semeval-2015 task 1: Paraphrase and semantic similarity in twitter (pit). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Denver, Colorado, pp. 1–11. Association for Computational Linguistics.

Yin, W. and H. Schütze (2015). Convolutional neural network for paraphrase identification. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pp. 901–911.