# Transfer Learning: From a Translation Model to a Dense Sentence Representation with Application to Paraphrase Detection

**Max K. Ferguson**
Department of Civil and Environmental Engineering
Stanford University
Stanford, CA 94305
maxferg@stanford.edu

## Abstract

It is becoming increasingly difficult to judge the originality of digital content, especially with the continuously growing corpus of digital text. Traditional methods such as n-gram overlap are susceptible to simple obfuscation techniques. In this paper, we investigate several methods of measuring sentence similarity, with particular emphasis on paraphrase detection. We show that a bag-of-vectors approach provides a simple yet effective baseline. We then demonstrate how the encoder from a neural machine translation model can be used to build a powerful paraphrase detection classifier.

Duplication of digital text, whether accidentally or deliberately copied, is a major concern for the academic and business communities, alike. Traditional plagiarism detection methods, such as n-gram overlap, are susceptible to simple obfuscation techniques (Potthast et al., 2010). A more robust method for comparing the semantic content of sentences is needed. Recently, researchers have shown that using dense representations of words and sentences, along with deep learning techniques, can provide impressive results on a number of natural language processing (NLP) tasks (Socher et al., 2011; Wu et al., 2016).

A major roadblock in the development of paraphrase classification models is the lack of large, high-quality datasets. Most of the large paraphrase datasets have less than 20,000 sentence pairs (Dolan and Brockett, 2005; Xu et al., 2014). While sufficient for traditional machine learning techniques, the publicly available paraphrase datasets are not sufficiently large to train complex deep learning models. The training difficulty arises because accurate paraphrase detection requires a certain level of semantic understanding. One solution is to apply transfer learning, where a natural language model is pretrained on a separate task, before being applied to the target task (Pan and Yang, 2010).

Transfer learning has seen many applications in image processing, where the lower layers of a pretrained model are often used to initialize a new model (Oquab et al., 2014). In some cases the weights of the lower layers can be fixed while training the second model. A similar approach is applied in (Pan et al., 2010) where a denoising autoencoder is trained on a large dataset and then used for a sentiment classification task.

In this work, we demonstrate how the encoder from a neural machine translation (NMT) model can be used to build a powerful paraphrase detection classifier. We start by constructing a simple bag-of-words baseline for the paraphrase detection task. We then demonstrate that the context vector from a translation vector can be used to classify paraphrases with greater accuracy than the baseline. Finally, we combine the baseline model with the NMT model, to generate an ensemble model which approaches state-of-the-art performance.

# 1 Background

Plagiarism detection has long been a topic of interest for both the academic and commercial communities. Traditional methods for detecting text similarity include substring-matching, fingerprinting and vector-similarity. In substring matching, the comparison of two text chunks is based on word-level n-grams (Barrón-Cedeño and Rosso, 2009). In fingerprinting, a block of text is hashed to form a fingerprint, which can be used to search for exact text matches. Vector methods attempt to form a *distributed representation* of text chunks, and use a distance metric like *euclidean distance* to measure similarity.

The use of distributed representations in natural language processing (NLP) has been largely fueled by the successful use of word vectors in a range of NLP tasks. Distributed representations of words in a vector space help learning algorithms to achieve better performance in NLP tasks by grouping similar words (Mikolov et al., 2013).

A number of researchers have attempted to develop distributed sentence representations including (Le and Mikolov, 2014) which attempted to create distributed representations of sentences and paragraphs. Although distributed sentence representations have been used to achieve state-of-the-art performance on a number of NLP tasks, there seems to be a lack of consensus in the generation of these representations. One of the major issues is that distributed representation comes more naturally to words than to sentences. While the semantic content of a word is defined by the context of the word, the semantic content of a sentence is defined by the interactions of its constituent words.

Nonetheless, the use of distributed representations has fueled recent advances in NMT, amongst other NLP fields. Researchers at Google recently demonstrated how they achieve zero-shot translation between unseen language pairs using a sequence-to-sequence model, with an intermediate representation that is common across all language pairs. We hypothesize, and subsequently demonstrate, that the intermediate layers of a NMT model can provide useful distributed sentence representations.

## 1.1 Datasets

In recent years, paraphrase detection has become a benchmark problem for a range of natural language models. Most researchers choose to benchmark their models on either the Microsoft Research Paraphrase Corpus (MSRPC) (Dolan and Brockett, 2005) or the Twitter Paraphrase Corpus (TPC) (Xu et al., 2014). The MSRPC contains 5801 sentences pairs, 66% of which are paraphrase pairs (Stein and zu Eissen, 2006). The same dataset has been arbitrarily split into a training set containing 4076 examples and a test set containing 1725 examples, by the original publishers.

There have been a number of studies on the MSRPC, including (Cheng and Kartsaklis, 2015) which obtained state-of-the-art results with a modified deep recurrent neural net (RNN) model, and (Socher et al., 2011) which obtained promising results with a deep recursive autoencoder. For the remainder of this paper, we consider the MSRPC as our primary test set for paraphrase detection.

# 2 Approach

In this work, we develop three models for the paraphrase detection task. The first is a simple baseline using pretrained GloVe vectors along with a shallow feedforward Siamese network. In the primary approach we train a translation model and use the context vector as distributed sentence representation for the paraphrase task. We conclude by developing an ensemble method that encompasses elements from the baseline and the primary model.

## 2.1 Baseline

For the baseline, we construct a single-layer feedforward Siamese neural network as shown in Figure 1. We represent each token in the dataset using 50-dimensional GloVe word vectors. Distributed sentence representations are created by averaging the word vectors in each sentence. The sentence vectors are passed through a single-layer neural network, before being compared using the cosine similarity measure. Each side of the Siamese network is constructed according to:

$$\boldsymbol{x} = \frac{1}{N_{words}} \sum_{i}^{N_{words}} \boldsymbol{e}_i$$

$$\boldsymbol{u} = \sigma(W\boldsymbol{x} + \boldsymbol{b})$$

where $\boldsymbol{e}_i$ is the GloVe embedding of word $i$, $\boldsymbol{x}$ is the distributed sentence representation, and $W \in \mathbb{R}^{20\times 50}$ is the weight matrix. The cosine similarity is used to compare the output of each neural network:

$$s = \frac{\boldsymbol{u} \cdot \boldsymbol{v}}{||\boldsymbol{u}|| \cdot ||\boldsymbol{v}||}$$

where $\boldsymbol{u} \in R^{20}$ is the output of the left neural network, $\boldsymbol{v} \in R^{20}$ is the output of the right neural network, and $s$ is the similarity score. The weight matrix is initialized to the identity matrix. The model is trained using the contrastive loss function:

$$loss(\boldsymbol{\theta}) = y(\max(0, m - d))^2 + (1 - y)d$$

where $y$ is the true label, $d$ is the cosine distance $(1 - s)$, and $m$ is the margin. This loss function encourages similar pairs to be at the same position, and penalizes dissimilar pairs which are closer than the margin.

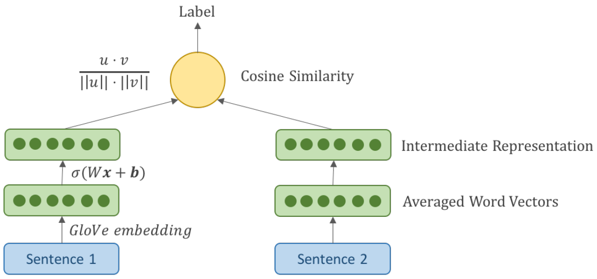
Label
$\frac{u \cdot v}{||u|| \cdot ||v||}$
Cosine Similarity
Intermediate Representation
$\sigma(W\boldsymbol{x} + \boldsymbol{b})$
Averaged Word Vectors
GloVe embedding
Sentence 1
Sentence 2

Figure 1: Siamese neural network with a feedforward neural encoder

### 2.2 Translation Model

The primary model is a sequence-to-sequence model that is trained as a translation model, but used to generate distributed representations of English sentences. This approach was inspired by recent advances in machine translation (Wu et al., 2016) as well as the use of recursive autoencoders (RAE) for paraphrase detection (Socher et al., 2011). In most cases, the context vector of a sequence-sequence machine translation model contains sufficient lexical and semantic information to fully reconstruct a sentence in another language. Therefore, we hypothesize that comparing two context vectors could provide a useful measure of similarity. To further this point, consider the case where two English sentences translate to the same French sentence; the English sentences are most likely paraphrases.

A recurrent neural network (RNN) with gated recurrent unit (GRU) cells is used for the encoder and decoder of the sequence-to-sequence translation model. A similar model has shown to be very effective for machine translation tasks (Cho et al., 2014). In the single-layer case, the RNN encoder is defined by the following equations:

$$\boldsymbol{z}_t = \sigma(W^{(z)}\boldsymbol{x}_t + U^{(z)}\boldsymbol{h}_{t-1})$$

$$\boldsymbol{r}_t = \sigma(W^{(r)}\boldsymbol{x}_t + U^{(r)}\boldsymbol{h}_{t-1})$$

$$\tilde{\boldsymbol{h}}_t = \text{ReLU}(r_t \circ U\boldsymbol{h}_{t-1} + W\boldsymbol{x}_t)$$

$$\boldsymbol{h}_t = z_t \circ \boldsymbol{h}_{t-1} + (1 - z_t) \circ \tilde{\boldsymbol{h}}_t$$

where $\boldsymbol{x}_t \in \mathbb{R}^{500}$ is the embedding for the input token at timestep $t$, and $\boldsymbol{h}_t$ is the output of the GRU. The decoder is defined by a similar set of equations, but each layer also receives the previous prediction as an input. The number of layers in the RNN is treated as a model hyperparameter.

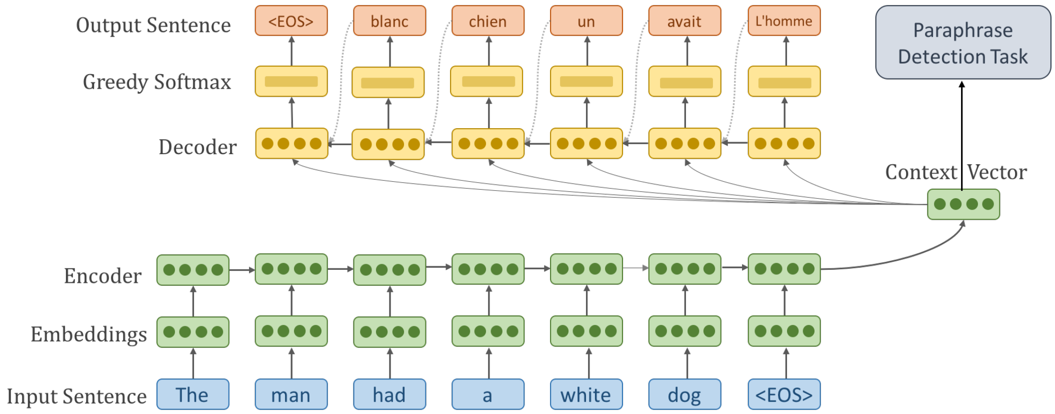

Figure 2: A RNN sequence-to-sequence model with single encoder and decoder layers. Note that the diagram also indicates how the context vector can be extracted and used for paraphrase classification

Figure 2 demonstrates the layout of the sequence-to-sequence model. We choose not to include an attention mechanism in the model, as it would allow information to bypass the context vector. The weights for the encoder and decoder are trained separately, as recommended in (Cho et al., 2014). The decoder is designed to produce a probability distribution for the next word, over the entire vocabulary. A greedy softmax function is used model the probability that token $x_j$ appears at timestep $t$:

$$p(x_{t,j}|x_{t-1},...x_t) = \frac{\exp(\boldsymbol{w}_j \boldsymbol{h}_t)}{\sum(_{j'=1}^{K} \exp \boldsymbol{w}'_j \boldsymbol{h}_t)}$$

for all possible tokens $j = 1, ..., K$. Note, $\boldsymbol{w}_j$ is the row of the softmax weight matrix corresponding to token $j$. In practice, we normalize the softmax using a random subsample of the vocabulary, to avoid iterating over the entire vocabulary. In order to obtain good translation results within a reasonable amount of training time, we use the following model configuration:

- Gated recurrent unit (GRU) cells in the encoder and decoder
- Between 1-4 hidden layers, each with 1024 cells
- Gradient clipping based on global gradient norm
- Enforced learning rate decay when dev accuracy plateaus
- Bucketing of input sentences based on length
- Reversed output (the decoder returns the output sentence in reverse order)
- Addition of EOS and PAD symbols to input sentences
- English and French vocabulary limited to most common 40K tokens
- Stochastic gradient descent optimizer

### 2.3 Transfer Learning for Paraphrase Detection

The encoder from the NMT model is used to generate distributed sentence representations for the paraphrase classification task. This allows the paraphrase classification model to generalize beyond the paraphrase training set, using knowledge stored in the NMT encoder. To form the distributed representation we concatenate the output from each layer in the encoder:

$$\boldsymbol{c} = \begin{bmatrix} \boldsymbol{h}^{(0)} \\ \vdots \\ \boldsymbol{h}^{(N)} \end{bmatrix}$$

where $\boldsymbol{h}^{(i)}$ is the output of encoder layer $i$ at the last time step. We then constrain the size of the context vector by adding $l2$ regularization:

$$\tilde{J}(\boldsymbol{\theta}) = J(\boldsymbol{\theta}) + \lambda \left\|\boldsymbol{c}\right\|_2^2$$

where $J(\boldsymbol{\theta})$ is the standard cross-entropy loss over the outputs and $\lambda$ is a model hyperparameter governing the context vector regularization strength.

The model can be used to predict the similarity of two English sentences. Assume $\boldsymbol{u}$ is the context vector for the first sentence and $\boldsymbol{v}$ is the context vector for the second sentence. The similarity $y_{\boldsymbol{uv}}$ of the sentences can be estimated using the cosine similarity:

$$y_{\boldsymbol{uv}} = \frac{\boldsymbol{u} \cdot \boldsymbol{v}}{||\boldsymbol{u}|| \cdot ||\boldsymbol{v}||}$$

The similarity metric $y_{\boldsymbol{uv}}$, can be converted to a binary label by comparing it to some threshold. In the simple case, we just choose the threshold to maximize accuracy on the paraphrase training set.

### 2.4 Ensemble

In this section, we discuss how the baseline and translation RNN approach are combined to create an Ensemble model that approaches state-of-the-art performance. Preliminary tests on the translation encoder reveal that the model is more sensitive to sentence structure than word choice. For example, the sentence “the man had a white dog” and “‘the man had a red dog” have a very high similarity score. Similarly, the RNN model labels the sentences “There are 5 apples in the basket” and “There are 9 apples in the basket” as identical. To overcome these issues we choose to combine the NMT model with the baseline model and six manually selected features, to create an ensemble model.

In both the GloVe and RNN models, numbers have very similar representations, but even small numeric differences are considered sufficient to reject a paraphrase relation in the MSRPC dataset. Therefore, we add three number features, that are commonly used with the MSRPC dataset (Bowman et al., 2015; Socher et al., 2011). The first is a binary feature which is 1 if the sentence pair contains exactly the same numbers or no number, and 0 otherwise. The second is 1 if the sentence pair contains the same numbers, and the third is 1 if the set of numbers in one sentence is a strict subset of the numbers in the other sentence. Since our GloVe model cannot capture sentence length we also add the difference in sentence length. As neither the GloVe model or the RNN can capture the number of exact string matches, we add the percentage of words and phrases in one sentence that are in the other sentence and vice-versa.

For the ensemble model we use these six features along with the similarity metric from the GloVe and RNN models. We train an $l2$-regularized logistic regression classifier on the MSRPC training set, and tune the regularization strength using cross-validation.

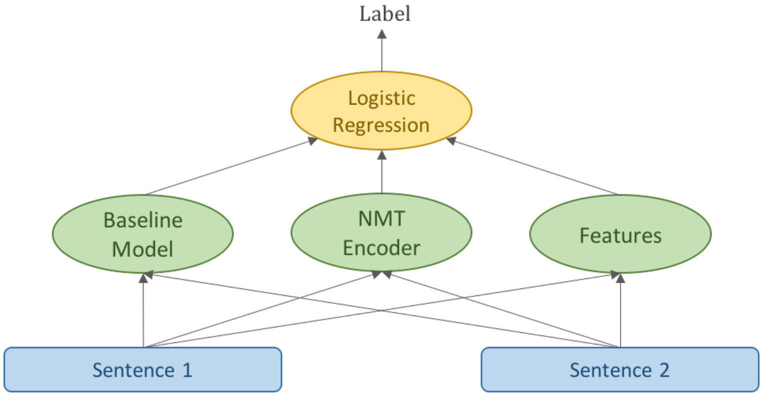

Figure 3: Ensemble model capturing information from GloVe vectors and NMT encoder.

## 3 Experiments

The baseline Siamese network and word embeddings were trained on an Amazon GPU instance, using the MSRPC dataset. The F1 metric is used to evaluate the performance of each model, as the labels in the MSRPC are unbalanced. An F1 score of 0.81 is obtained using the baseline, as shown in Table 2.

The translation model is trained on the WMT '14 Gigaword English-French dataset using the modified loss function $\tilde{J}(\boldsymbol{\theta})$. Originally, we alternated between training the model on the paraphrase and translation tasks. However, the additional complexity of the training code did not justify the small accuracy gains that we observed.

The NMT model allows us to compare arbitrary sentence pairs using the cosine similarity. Table 1 shows some examples of the similarity scores assigned to various sentence pairs. The first two examples in Table 1 are clearly paraphrases, and are labeled with high similar scores by the NMT model. The third example compares two sentences formed with the words from "Only she told me that she loved me". Interestingly, the bag-of-words baseline would label these two sentences as identical, but the NMT correctly assigns a low similarity score of 0.41.

An alternative way of analyzing the NMT context vectors is shifting them into two-dimensional space using principal component analysis (PCA). Figure 4 shows how the NMT model clusters sentences by meaning. The exception is that the sentences in the form "There are *x* apples in the basket" are clustered together. While these sentences are semantically and lexically similar, they are not paraphrases by the MSRPC definition.

Table 2 demonstrates how the models in this paper compare to the current state-of-the-art. The NMT model surpasses the baseline by a relatively large margin. It also surpasses the previous state-of-the-art for an unsupervised algorithm.

Table 1: Cosine similarity comparison of context vectors for various different sentences.

| **Sentence 1** | **Sentence 2** | **Similarity** |
|---|---|---|
| The man had a white dog. | The man had a dog that was white. | 0.97 |
| I'm never going to forget this day. | I am not going to forget this in my life. | 0.81 |
| *Only* she told me that she loved me. | She told me that she loved *only* me. | 0.45 |
| They're in the garden. | There in the garden. | 0.41 |
| Two people in the car. | Too many people in the car. | 0.23 |
| She was very cold. | The climb was long and difficult. | 0.18 |

Table 2: Comparison of model performance on MSRPC paraphrase prediction task

| **Classifier** | **Accuracy** | **F1** |
|---|---|---|
| GloVe bag-of-words baseline | 71.0% | 80.1% |
| GloVe bag-of-words baseline + feats | 73.4% | 81.9% |
| Neural machine translation encoder | 72.0% | 81.6% |
| Neural machine translation encoder + feats | 75.3% | 82.7% |
| Ensemble model + feats | 76.3% | 83.2% |
| WordNet similarity (unsupervised) (Fernando and Stevenson,2008) | 74.1% | 82.4% |
| Recursive autoencoder + feats (Socher et. al., 2011) | 76.8% | 83.6% |
| Syntax-aware recursive NN (Cheng and Kartsaklis, 2015) | **78.6%** | **85.3%** |

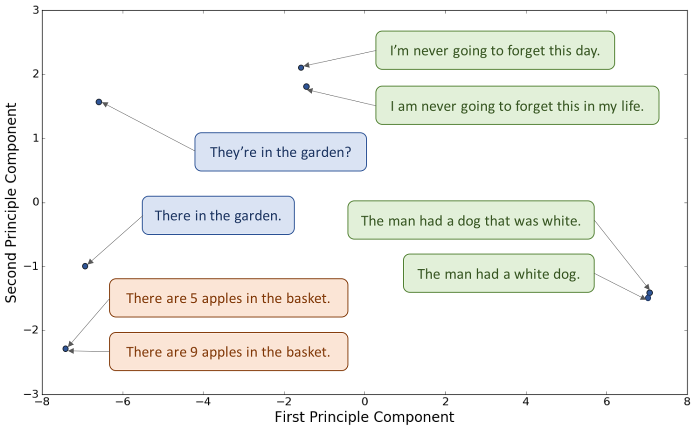

Figure 4: Visualization of NMT context vector using PCA. Note that the two paraphrase pairs (green) are correctly grouped together, while the red pair is incorrectly grouped together.

## 4 Discussion

Deep learning provides a powerful method for extracting syntactic and semantic and relationships from text. However, training modern deep learning models requires large and relatively high quality datasets. Transfer learning provides a way to transfer knowledge gained from one particular task to another. In many cases, it can be used to improve the generalizability of a model to a particular task, especially when there is only a small amount of data available for the target task.

In this paper, we demonstrated that a translation model can be used to encode sentences into a dense vector form, while capturing relevant lexical and semantic information. The distributed sentence representations provided useful in a paraphrase classification task. Unlike the baseline, the NMT encoder was able to capture syntactic relationships between words. However, the NMT was unable to differentiate sentences based on numeric values, or similar subtleties in semantic content. To overcome this problem, an ensemble model was trained using the bag-of-words baseline, the NMT model, and a small set of features. The ensemble model achieved commendable performance on the MSRPC dataset.

The NMT model was able to reach state-of-the-art performance on the MSRPC for an unsupervised algorithm. Undoubtedly, an accuracy gain could be achieved by jointly training the model on both the paraphrase and the translation tasks. However, we found hyperparameter selection very difficult during the joint training process. With the wrong hyperparameters the NMT model quickly overfitted or underfitted the MSRPC dataset. Training the NMT model alone took 1-2 days on a K80 GPU, greatly restricting manual hyperparameter trialling. However, we still believe that joint training with a translation task could provide useful for a range of NLP tasks.

## 5 Future Work

There are a range of possibilities for future work following this work. The first is to explore joint training with automatic hyperparameter search. In this scenario, the hyperparameters in the loss function would be updated at each epoch to ensure that the model was fitting well to both tasks. For example, if the model started to overfit the paraphrase task, then the the gradient updates would become more constrained on the next epoch.

It would also be beneficial to explore NMT transfer learning with other datasets or NLP tasks. The MSRPC is a relatively small and noisey dataset; it would be interesting to see how the model performed on several different paraphrase corpus's. Finally, it would be interesting to see if NLP tasks could be generalized across languages using transfer learning with a multi-language NMT model.

### Acknowledgments

Thanks are due to Prof. Christopher Manning, and Prof. Richard Socher for facilitating such a fantastic course. We also thank the course assistants and teaching staff for their assistance throughout the course, and their guidance during the final project. Finally, we would like to our project mentor, Abi See, for pointing us in the right direction from the start of the project.

### References

Alberto Barrón-Cedeño and Paolo Rosso. On automatic plagiarism detection based on n-grams comparison. In *European Conference on Information Retrieval*, pages 696–700. Springer, 2009.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

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

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

William B Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proc. of IWP*, 2005.

Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196, 2014.

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

Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web*, pages 751–760. ACM, 2010.

Martin Potthast, Benno Stein, Alberto Barrón-Cedeño, and Paolo Rosso. An evaluation framework for plagiarism detection. In *Proceedings of the 23rd international conference on computational linguistics: Posters*, pages 997–1005. Association for Computational Linguistics, 2010.

Richard Socher, Eric H Huang, Jeffrey Pennington, Andrew Y Ng, and Christopher D Manning. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *NIPS*, volume 24, pages 801–809, 2011.

Benno Stein and Sven Meyer zu Eissen. *Near Similarity Search and Plagiarism Analysis*, pages 430–437. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-31314-4. doi: 10.1007/3-540-31314-1_52. URL `http://dx.doi.org/10.1007/3-540-31314-1_52`.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Wei Xu, Alan Ritter, Chris Callison-Burch, William B Dolan, and Yangfeng Ji. Extracting lexically divergent paraphrases from twitter. *Transactions of the Association for Computational Linguistics*, 2:435–448, 2014.

## Appendix

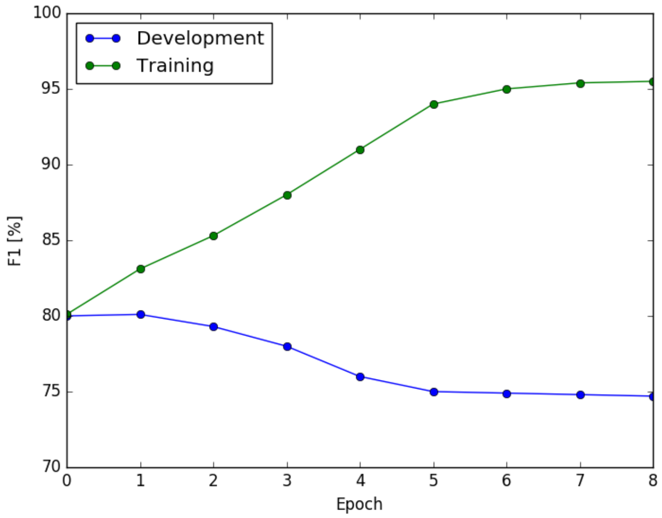

Figure 5: F1 score on the training and development sets, while training the the baseline model. When generating this figure we allowed the GloVe vectors to be trained, which caused the model to quickly overfit. In subsequent tests we fixed the GloVe vectors

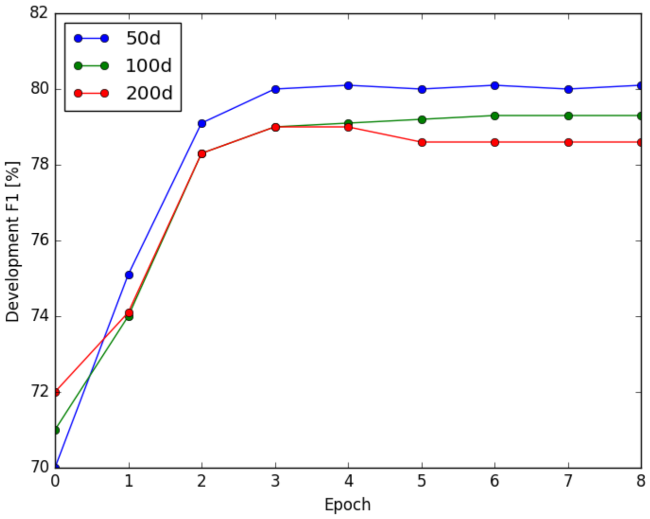

Figure 6: F1 score on the development set, while training the the baseline model with fixed GloVe vectors. In subsequent tests we chose to use 50d GloVe vectors

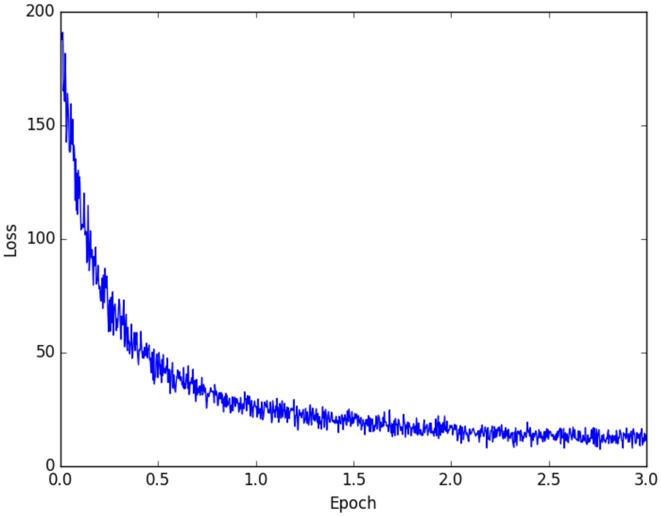

Figure 7: Training loss on the NMT translation model. Note, each epoch contains 1 billion tokens and took about 24 hours to run.

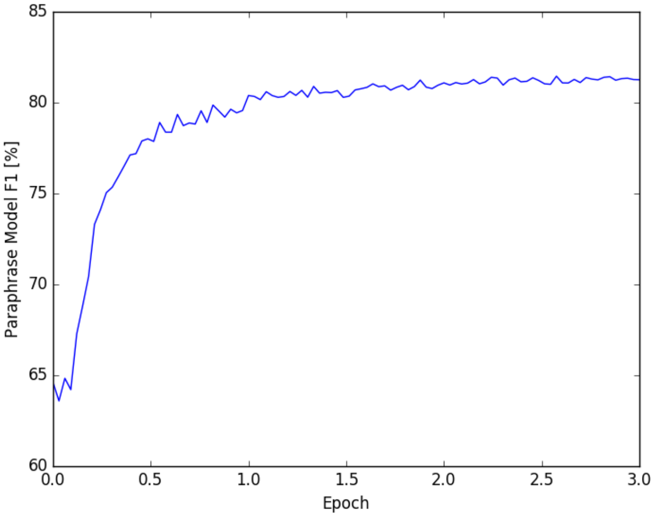

Figure 8: F1 score on the MSRPC dataset as the NMT model is trained. In this test there are no additional features added to the NMT paraphrase model