Size v. Specificity: Comparing Pre-trained Embeddings for Disaster Tweet Classification

Bill Pung Tuck Weng*
Interdisciplinary Graduate School
Nanyang Technological University
pung0013@e.ntu.edu.sg

Tan Xian Ren, Clement*
School of Computer Science and Engineering
Nanyang Technological University
S190099@e.ntu.edu.sg

Abstract

With the proliferation of social media, Twitter has become an important channel for emergency respondents to conduct automated monitoring of on-the-ground updates on disasters. To this end, Disaster Tweet Classification is a motivated task to identify if tweets are referencing disasters or not. We begin with training an LSTM encoder to learn the word embeddings from a corpus built using the "Real or Not? NLP with Disaster Tweets" classification dataset, with a final linear classifier layer for the task. However, the success of this model is limited, as the disaster tweet classification dataset available is small. With the advent of pre-trained models in NLP that achieves state-of-the-art (SOTA) results on many tasks [1, 2], we sought to leverage pre-trained embeddings as inputs to our classifier. In choosing the ideal corpus for pre-training, we observe a trade-off between size and domain-matching; increasing the size of a pre-trained corpus makes the source domain distribution less similar to the target domain. This brings us to the central question: Is it more effective to have a larger but more general pre-trained corpus, or a smaller one that is more similar to the target domain? Our experiments on LSTM with GloVe embeddings and pre-trained BERT show the former, highlighting the importance of large corpora when pre-training for low-resource transfer learning.

1 Introduction

Today, news spread rapidly through social networking platforms like Twitter and Facebook. This has led to the proliferation of online falsehoods (whether intentional or otherwise), which are especially damaging when propagating fear in public during times of distress, such as the advent of an pandemic like Covid-19. To combat this, we seek to build an effective means of detecting if a given tweet is referencing disasters. This is achieved by building a classifier which would predict if a given tweet is referencing disasters or otherwise, a non-trivial task that requires the model to look into the meaning of an entire sentence for topic detection. For instance, the tweet "Look at the sky last night, it was ablaze" contains a word distributionally similar to disasters (i.e. ablaze), but takes on a metaphorical meaning denoting the colour red (See Figure 3 in Appendix).

We begin with a Long Short-Term Memory (LSTM)[3] model as an encoder, to learn the word embeddings on a corpus built using tweets from our target dataset. One natural drawback of this method is the small size of the dataset, that limits the training signal provided to the encoder in learning the distributions of words; this results in a model with a test F1 score of 71.5%. A natural next step is to use pre-trained embeddings, to leverage transfer learning in improving the word representations provided to the classifier.

The trade-off encountered when choosing pre-trained embeddings now comes into play- do we choose embeddings pre-trained on a larger but more general corpus, or a smaller one that is more similar to the target domain? This question is progressively pertinent, as leveraging the SOTA models today potentially saves us many GPU-months of training on increasingly colossal corpora [1, 4, 5, 6, 2], and

*Both authors contributed equally to the project

equivalently over 600k lbs of CO_2 emissions [7]. On the other hand, corpora well-aligned with our target domain can be costly to procure, and possibly irrelevant with ever-evolving nature of natural language.

In this paper, we investigate the aforementioned trade-off, by comparing LSTM models with GloVe embeddings pre-trained on Wikipedia, Twitter and Common Crawl, with sizes of 6, 27 and 42 billion tokens respectively. We note that increasing the size of the pre-trained corpus leads to increases of $2-3\%$ F1 score over the baseline LSTM on our classification task. We then run the experiments on BERT [1], obtaining a further increase of 2.6% over the best LSTM model, for a final test F1 score of 77.2%.

Next, we analyse corpora out-of-vocabulary (OOV) statistics to suggest an explanation for the increase in performance, and surmise that an increase in corpus size reduces the number of OOV tokens, directly improving the results of our downstream classification task.

We summarize the key contributions in our paper:

- We investigate the trade-off between size and domain-matching when using pre-trained embeddings in transfer learning, and show that size dominates domain-specificity;
- We analyse the coverage of pre-trained word embeddings, and show that a larger pre-trained corpora has fewer Out of Vocabulary tokens, improving classifier performance;
- We empirically verify the efficacy of transfer learning on low-resource setting on a tweet veracity task, Disaster Tweet Classification.

2 Related Work

With rapid advancements in the field of deep learning, SOTA models have achieved near-human performance on a plethora of NLP tasks [8, 4, 5, 6, 2, 9, 10]. One such task is disaster tweet classification. Deep learning architectures are able to utilize pre-trained word embeddings such as GloVe and Word2Vec [11, 12] to learn distributed word representations.

Ngyuen et al. [13] show the efficacy of using neural networks in out-of-domain disaster tweet classification over machine learning methods, such as Support Vector Machines, Random Forests and Logistic Regression. In [11], they made comparisons between Crisis Word2Vec [14] and GloVe [15] embeddings that were trained on the CrisisNLP dataset. Furthermore, they noted that there was no need to use domain-specific word embeddings in improving performance. However, we note that from [16], there are inconsistencies in performance between GloVe and Word2Vec, as performance is dependent on the application domain. Our work differs from [11], as we use an LSTM and BERT, and leveraged different GloVe embeddings such as GloVe.6B, GloVe.Twitter.27B as well as GloVe.42B on a smaller dataset - "Real or Not? NLP with Disaster Tweets".

3 Approach

We use the LSTM as a baseline, serving as an encoder to generate a hidden representation of each input sequence that is then fed into a linear layer for binary classification. In addition, different word tokenization techniques and embeddings have been incorporated into the network to determine which pre-trained embeddings set is suited for this task. We optimize the models by minimizing the cross-entropy loss.

3.1 LSTM Baseline

For the first baseline, we use an LSTM as a model to learn the word embeddings, feed the final hidden representation of the LSTM encoder into a fully connected linear classifier to produce the probability over two output classes (disaster or not).

3.2 GloVe

With a train corpus of 74k tokens, the dataset is rather small, which may not allow efficient learning of word embeddings from scratch. An area of improvement is using pretrained word embeddings. To

this end, we incorporated the use of GloVe [15] in the LSTM model, using the version pre-trained on a combined corpus sourced from Wikipedia 2014 and Gigaword 5, totalling 6B tokens. This version has a vocabulary size of 400K, with each token being represented by a 300-dimensional vector. In addition, we also include the use of GloVe Common Crawl which has 42B tokens, 1.9M uncased vocab with 300-dimensional vectors.

We intialize the nn.Embedding layer, load the GloVe pre-trained weights and set the requires_grad parameter to False, thereby preventing the training of the embedding layer. Other parts of the network are left untouched.

3.3 GloVe Twitter

To further explore the advantages of using transfer learning on a small dataset closer to the disaster tweet classification task, we sought pre-trained embeddings trained on a domain closer to our target domain. Speficially, we used word embeddings pre-trained on Twitter data. The next improvement included the usage of GloVe pre-trained on a Twitter corpus of 2B tweets comprising 27B tokens, producing a vocabulary size of 1.2M tokens of 200-dimensions each.

3.4 BERT

We used the HuggingFace [17] implementation of the BERT-Base Model [1]. In this model, the train and test corpus are loaded and tokenized using WordPiece embeddings [18] of a 30K token sized vocabulary. To generate the embeddings, the model was pre-trained on the BookCorpus of 800M words, and English Wikipedia of 2.5B words. Each sequence comprises a single sentence that is prefixed and suffixed with a classification token ([CLS]) and an end-of-sentence token ([SEP]) respectively, and finally padded (with a [pad] token) until a maximum length of 64 tokens. The final hidden state corresponding to the [CLS] token is fed into a linear layer for binary classification.

4 Experiments

4.1 Data

4.1.1 Exploring the Dataset

The dataset that we use is the "Real or Not? NLP with Disaster Tweets" dataset found on Kaggle [2]. The dataset contains $10,000$ tweets that were manually annotated. The dataset is split into train and test csv files. The train file contains 5 columns: id, keyword, location, text and target while the test file contains 4 columns: id, keyword, location and text.

4.1.2 Data Pre-Processing

Before pre-processing the train and test data, it is good to have an idea of what needs to be done. Visualizing the different tweets in the train dataset (refer to Figure 1), we can see that there are hashtags, contractions, emojis, Mojibake (the result of using an unintended character encoding), punctuation, hyperlinks, capitalized letters, words and many others. It is important to normalize these words into tokens such that words like "Cat" and "cat" are the same.

First, we imported a list of contractions from [3] , which expands contractions like "ain't" to "am not", "aren't" to "are not" etc. As we need to remove punctuation, words like "aint" would require processing and should not be used. That is why there is a need to expand these contractions before other pre-processing steps. After the contractions are removed, we proceed to lower the case all tweets, removing punctuation, hyperlinks, hashtags (but keeping the words), and Mojibake. Next, we remove stop words like "i", "the", "we", etc. Once the dataset has been cleaned, we built dictionaries of words found in the train and test datasets.

Lastly, we built a tokenizer from scratch, by simply assigning an index to the word by going through the entire dataset, tweet by tweet. While assigning indices to tokens, we built word-to-index and index-to-words dictionaries to facilitate efficient lookup. After pre-processing the sentences into

[2]Obtained from https://www.kaggle.com/c/nlp-getting-started

[3]http://stackoverflow.com/questions/19790188/expanding-english-language-contractions-in-python

id	keyword	location	text	target
884	bioterrorism		Firepower in the lab [electronic resource] : automation in the fight against infectious diseases and bioterrorism /Â‰Ã›_ http://t.co/KvpbybglSR	0
885	bioterrorism		@CAgov If 90BLKs&8WHTs colluded 2 take WHT F @USAgov AUTH Hostage&2 make her look BLK w/Bioterrorism&use her lgl/org IDis ID still hers?@VP	1
886	bioterrorism		@DarrellIssa Does that 'great Iran deal' cover bioterrorism? You got cut off terrible of them. Keep up the good work.	1
888	bioterrorism	San Francisco, CA	A Tale of Two Pox - Body Horrors http://t.co/W2IXT1k0AB #virus #infectiousdiseases #bioterrorism	1
960	blaze	Durham N.C	@GuiltyGearXXACP yeah I know but blaze blue dont have a twitter lol I drew this a few weeks ago http://t.co/sk3I74FLzZ	0
961	blaze	Delhi	#socialmedia news - New Facebook Page Features Seek to Help Personalize the Customer Experience http://t.co/nbizaTlsmV	0
962	blaze	ARIZONA	@DJJOHNBLazE shout out blaze the hottest DJ in the Sothwest	0
4100	drought		U.S. in record hurricane drought http://t.co/8JvQI9UspL	1
4492	electrocuted		Elsa is gonna end up getting electrocuted. She's gonna end up like that cat from christmas vacation.	0
4494	electrocuted		Not being able to touch anything or anyone in Penneys without being electrocuted ??	0

Figure 1: Example of tweets from the dataset

tokens, we have a total of 74144 tokens in the corpus, of which there are 15560 unique tokens in the vocabulary. A visualization of the word cloud is shown, with the size of the words proportional to the frequency in the dataset (Figure 2).

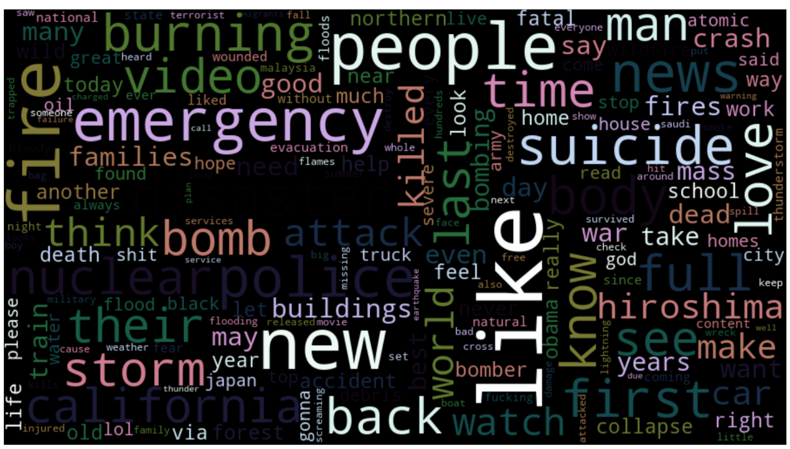

Figure 2: Word cloud of the disaster tweet classifcation after cleaning

4.1.3 Dataloaders

Before feeding batches of tweets into the model, we have to ensure that the tweets are of equal length. Therefore, the maximum sequence length of tweets is set to 21 words for the LSTM. As BERT uses WordPiece tokenization to further break words into their constituents, we use a maximum sequence length of 40 tokens for complete coverage. Tweets that fall short of the max sequence length are padded with zeros for the LSTM model while a "[pad]" token is used for padding by the BERT tokenizer. We use the Dataset and Dataloader classes from PyTorch to efficiently batch our inputs and provide corresponding ground truth labels for the classification task.

In addition, we employ the following packages and libraries: TorchText for pre-processing, Pandas, Numpy, popular datasets for NLP, and pre-trained word embeddings such as Global Vectors for Word Representation (GloVe) [15].

4.2 Evaluation method

The goal of the task is to predict whether a given tweet is a real disaster or not. Hence, a natural evaluation metric to use for this task is the F1 score. For our implementation, we used the F1 score function provided by sklearn, and report the score over the test dataset. To ensure that there is no overfitting, we visualized the training and test data losses and perform early stopping when necessary.

4.3 Experimental details

The experiments were done using the HuggingFace Transformer Library [17] on PyTorch. We utilized NVIDIA 2080 Ti GPUs to run our models. For all four configurations, we trained the model with the Adam optimizer with a batch size of 32. The learning rates of 1e-3, 5e-4 and 2e-5 were used for the LSTM-Learned, LSTM-GloVe and BERT models respectively, which were found to produce the best test results from a grid-search conducted over the hyperparameters.

4.4 Results

Model	Total Params	Embedding	Train Accuracy	Test Accuracy	Test F1
LSTM	4.51M	Learned	**84.653** (2.49)	76.574 (1.08)	71.503 (1.00)
LSTM	1.03M	GloVe.6B	82.396 (1.82)	78.688 (0.48)	73.709 (0.44)
LSTM	0.773M	GloVe.Twitter.27B	81.142 (1.01)	78.749 (0.36)	74.215 (0.27)
LSTM	1.03M	GloVe.42B	82.071 (0.84)	**79.105** (0.61)	**74.615** (0.46)
BERT	109M	WordPiece	**85.633** (2.18)	**81.024** (0.51)	**77.218** (0.29)

Table 1: Sentence Classification accuracy and F1 scores (in %) of models with different embeddings. We report the average scores over 5 runs, and the corresponding standard deviation in parenthesis.

From table 1, we see substantial improvement in using pre-trained word embeddings over training our own embeddings from scratch, with improvements of $2-3\%$ in test F1 score over the baseline LSTM model. We surmise that this is because the original corpus is too small to learn fine-grained representations of the words.

Contrasting GloVe embeddings pre-trained on 6 billion tokens from Wikipedia 2014 & Gigaword-5 with embeddings pre-trained on 27 billion tokens from Twitter, we see a reasonable improvement of the latter by 0.6%. When using GloVe embeddings pre-trained on 42 billion tokens from Common Crawl, we see a further improvement over the Twitter embeddings by 0.4%. One clear trend we observe is that increasing the pre-trained corpora size from 6 to 42 billion tokens increases the test performance of the classifier. This increase surpasses the benefits gained when using embeddings pre-trained on a more specific corpus (i.e. GloVe.Twitter.27B)- we shed more light on this phenomena in section 5.

Using the BERT-Base model, we observe the best results in training accuracy, test accuracy and F1 scores. On one hand, it is surprising that the LSTM model using Twitter pre-trained emebddings would not fare as well as the BERT model pre-trained on more general corpora, as the former would be better poised to match the target distribution of Twitter words. However, we note that the BERT model is two orders of magnitude larger at 109 million parameters, and we surmise that this allows BERT to perform significantly better than the smaller LSTM models.

5 Analysis

To probe further into why a larger pre-training corpus size allows for better downstream performance, we investigate our conjecture that a larger pre-trained corpus provides better vocabulary coverage over the target domain. We look at the number of OOV tokens in the dataset (comprising the train and test sets), and the number of tweets in the dataset containing OOV tokens. Note that this analysis does not apply to BERT, which relies on WordPiece embeddings to attain 100% coverage of tokens in the dataset; we only look at LSTM models to isolate the effects of varying embeddings, controlling for model size.

From table 2, we see that increasing the pre-trained corpus size leads to a reduction in OOV tokens. While there is a large OOV vocabulary size of 20% and 13% for GloVe 6B and 42B respectively, this

Model	Embedding	Source vocab size	Dimension	OOV vocab size	OOV tweets
LSTM	GloVe.6B	0.4M	300	3110	3153
LSTM	GloVe.Twitter.27B	1.2M	200	2971	3267
LSTM	GloVe.42B	1.9M	300	2153	2308

Table 2: Pre-trained word embeddings statistics showing the number of tokens in the source vocabulary, unique Out of Vocabulary (OOV) tokens in the target vocabulary, and the number of tweets containing at least one OOV token. Note that the total number of words in the target vocabulary is 15560, while the total number of tweets in the target train and test set is 10876.

may not be surprising, as the words used in the Twitter domain might be sufficiently different from more general corpora such as Wikipedia. However, we note that despite being pre-trained on Twitter, the GloVe.27B embeddings could not cover 2971 out of 15560 words, essentially missing out 19.1% of the words, causing 30% of the dataset to contain at least one OOV token.

We surmise that while the Twitter embeddings are more aligned in domain with our target vocabulary, its source vocabulary is smaller than the GloVe.42B embeddings, naturally accounting for the shortfall in target vocabulary matching. This inherent trade-off in vocabulary size and efficacy of learned embeddings highlight the importance of having a large pre-trained corpus when generating word embeddings: The desiderata of smaller OOV vocabulary arises from having a larger source vocabulary, which must come from a corresponding increase in pre-trained corpus size, *ceteris paribus*.

6 Future Work

One avenue to increase points of comparison is to use alternate forms of word embeddings, e.g. FastText [19] or word2vec embeddings pre-trained on Twitter [14]. Given more computing resources, it would be interesting to investigate pre-training BERT from scratch on a Twitter dataset under the same conditions as [1]. This was beyond our scope of resources available to us and is thus left for future work.

7 Conclusion

To conclude, we show the benefits of incorporating pre-trained word embeddings to boost the classification performance of our models. We also noted a slight performance increase when we use word embeddings that share similar domain with the task. However, even though we used the GloVe Twitter embeddings, the LSTM model performed poorer when compared to larger GloVe embeddings and BERT. This suggests that using pre-trained word embeddings trained on a larger corpus provides better downstream performance than using domain specific word representations.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[2] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

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

[4] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[5] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

[6] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764, 2019.

[7] Zhanghao Wu, Zhijian Liu, Ji Lin, Yujun Lin, and Song Han. Lite transformer with long-short range attention. *arXiv preprint arXiv:2004.11886*, 2020.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[9] Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. Multimedia lab@ acl wnut ner shared task: Named entity recognition for twitter microposts using distributed word representations. In *Proceedings of the workshop on noisy user-generated text*, pages 146–153, 2015.

[10] Shuai Zhang, Lina Yao, and Aixin Sun. Deep learning based recommender system: A survey and new perspectives. *CoRR*, abs/1707.07435, 2017.

[11] Reem ALRashdi and Simon O'Keefe. Deep learning and word embeddings for tweet classification for crisis response. *arXiv preprint arXiv:1903.11024*, 2019.

[12] Md Yasin Kabir and Sanjay Madria. A deep learning approach for tweet classification and rescue scheduling for effective disaster management. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 269–278, 2019.

[13] Dat Tien Nguyen, Kamela Ali Al Mannai, Shafiq Joty, Hassan Sajjad, Muhammad Imran, and Prasenjit Mitra. Rapid classification of crisis-related data on social networks using convolutional neural networks. *arXiv preprint arXiv:1608.03902*, 2016.

[14] Muhammad Imran, Prasenjit Mitra, and Carlos Castillo. Twitter as a lifeline: Human-annotated twitter corpora for nlp of crisis-related messages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may 2016. European Language Resources Association (ELRA).

[15] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[16] Marwa Naili, Anja Habacha Chaibi, and Henda Hajjami Ben Ghezala. Comparative study of word embedding methods in topic segmentation. *Procedia computer science*, 112:340–349, 2017.

[17] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

[18] 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.

[19] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.

A Appendix

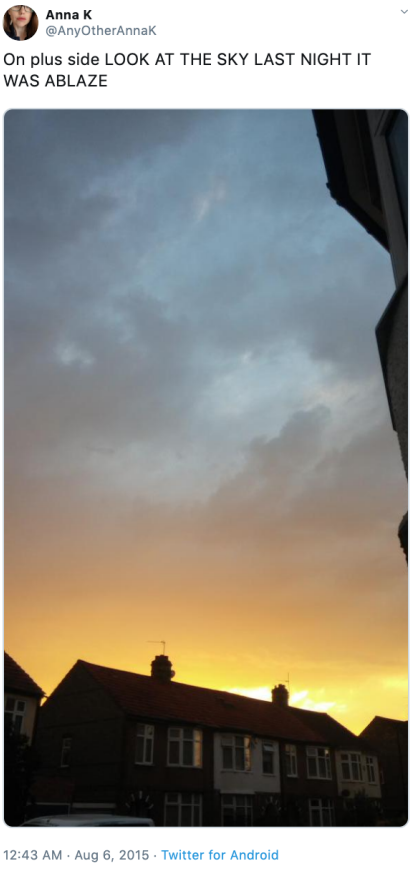

Figure 3: Example tweet exemplifying the complexity of the task. Ground truth label: Not about disaster (i.e. False)