2019-9

Estimating Distributed Representation Performance in Disaster-Related Social Media Classification

Pallavi Jain
Technological University Dublin, pallavi.jain@tudublin.ie

Robert J. Ross
Technological University Dublin, robert.ross@tudublin.ie

Bianca Schoen-Phelan
Technological University Dublin, bianca.phelan@tudublin.ie

Recommended Citation

Jain, P., Ross, R., Schoen-Phelan,B. (2019). Estimating distributed representation performance in disaster-related social media classificatio. *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*

2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

Estimating Distributed Representation Performance in Disaster-Related Social Media Classification

Pallavi Jain, Robert Ross, Bianca Schoen-Phelan
School of Computer Science
Technological University Dublin
Dublin, Ireland
{pallavi.jain, robert.ross, bianca.phelan}@dit.ie

***Abstract*—This paper examines the effectiveness of a range of pre-trained language representations in order to determine the informativeness and information type of social media in the event of natural or man-made disasters. Within the context of disaster tweet analysis, we aim to accurately analyse tweets while minimising both false positive and false negatives in the automated information analysis. The investigation is performed across a number of well known disaster-related twitter datasets. Models that are built from pre-trained word embeddings from Word2Vec, GloVe, ELMo and BERT are used for performance evaluation. Given the relative ubiquity of BERT as a standout language representation in recent times it was expected that BERT dominates results. However, results are more diverse, with classical Word2Vec and GloVe both displaying strong results. As part of the analysis, we discuss some challenges related to automated twitter analysis including the fine-tuning of language models to disaster-related scenarios.**

***Index Terms*—Text classification, Twitter, Word Embedding, ELMo, BERT**

I. Introduction

Twitter is a micro-blogging platform that has for many years enjoyed strong popularity. During natural and man-made disasters, Twitter has been used by lay-people and government officials alike to broadcast information [1], [2], [3].

During disaster situations, updates about the situational awareness comes in the form of text and images about locality damage, missing or trapped people or urgent needs. Example tweets for for example be:
"Due to rising water levels, Centre Street bridge has been closed."
These updates help the first responders and a decision making team in two ways: (i) understanding the situation around a specific area, and (ii) identifying urgent resource requirements. This information can be made more visible if provided in a classified manner, is expected to improve planning efficiency. Although crisis responses are issued on Twitter in near-real time, many tweets are actually not relevant to the events in question [4] and consequently need to be classified according to the informativeness of a particular tweet, and then in terms of information type [5], [6], [7].

A long-term posthoc analysis of tweets across the full social network can indicate the informativeness of a tweet due to re-tweets and social network structure. Real-time processing of social media in a disaster scenario means that early detection of informativeness is essential, and cannot be delayed for the benefit of complete social network analysis. Therefore, informativeness analysis reduces to a text classification task where the content of a tweet becomes essential in estimating its utility shortly after broadcast.

For tweet classification, natural language processing (NLP) offers many feature representation techniques that capture the structure of the text in the form of a numerical vector representation. While bag of word (BoW) or term frequency-inverse document frequency (TF-IDF) [8] were historically the most dominant representation types, recent times have seen a strong swing towards distributed representation commonly generated using variants of Deep Neural Network tasks [9]. One challenge with distributed representations, however, is that they tend to suffer from out-of-vocabulary issues [10]. This affects twitter classification tasks as tweets tend to consist of hashtags, slang words and abbreviations, which many models do not consider as part of the representation. Moreover, while distributed representations have been in existence for a number of years, the last two years have witnessed a move towards Transformer based architectures [11].

Representations underpin any classification task, and as such, it is essential that an appropriate representation is chosen to maximise potential performance in the disaster analysis scenario. Consequently, we performed a systematic assessment of the usefulness of different representation types in a tweet informativeness estimation task.

II. Related Work

A comprehensive body of research exists in relation to disaster analysis using NLP and image analysis in order to improve situation awareness. The research work in this area includes tweet classification [6], [7], [12], text summarisation [13], or multimodal approaches [14], [15] in order to have wider perspective of the disaster scenario. However, with respect to situation awareness, tweet classification has been

ASONAM '19, August 27-30, 2019, Vancouver, Canada

ACM ISBN 978-1-4503-6868-1/19/08...$15.00
https://doi.org/10.1145/3341161.3343680

the most researched area due to the availability of labelled data such as CrisisLex [16] and CrisisNLP [6].

Disaster tweet classification for the event type, informativeness, and information type is not new but several running tools like AIDR [5], Tweedr [2], CREES [17], and recently released CrisisDPS [12] has shown the continuous need of improvement in order to produce precise information. For example, CrisisDPS is one such tool where real-time tweet streams have been classified into three levels and have shown competitive results with the use of Word embedding and CNN.

As these tools utilise word embeddings for real-time classification and have achieved great performance, this study compares the latest state of the art in embeddings that is BERT and ELMo to traditional Word2Vec and GloVe embeddings.

A. Representations

For any social media classification task, the representation used plays a key role as it either enables or hinders certain types of analysis. Raw text-based representations such as BoW or TF-IDF have been very useful due to their transparency, but in recent years have been surpassed by a large collection of distributed representations that allow better generalisation than can be achieved with raw text-based representations. Word2Vec [18], Global Vector (GloVe) [19], and the recently proposed ELMo (Embeddings from Language Models) [20] and BERT (Bidirectional Encoder Representations from Transformers) [11] models are key examples of such distributional representations.

Word2Vec and GloVe are key instances of distributed representations for individual words. These models capture a semantic representation of an individual word, and the meaning of a complete sentence is captured as a vector of individual word vectors. These distributed word embeddings play an important role in text classification as they provide different aspects of the text; such as semantics similarity, syntactic relationships, and contextual relevance. Word2Vec converts the text corpus into a vector space model of similar words able to hold inner relationships between words in addition to a scalar distance. The GloVe is a bi-linear regression model that leverages both the traditional count of words approach and co-occurrence statistics from the Word2Vec approach.

In the last year, a number of more advanced representation types have been proposed to capture an increased level of context about language use in the representation. The two most prominent examples of such representations are ELMo [20] and BERT [11]. Both these models are built through deep contextualisation. ELMo leverages bi-directional LSTM and is trained on several language models, thus gaining the sense of words from preceding and succeeding word sequences. More specifically, ELMo captures the context and syntax of a sentence using vector representations, which are the linear function of all the input layers of the bidirectional language models (biLM). BERT meanwhile uses the concept of masked language models and next sentence prediction where it masks 15% of words randomly and runs them on a multi-layer bi-directional transformer encoder to keep the distributional contextual representation of words. After the masking process, it performs the next sentence prediction. Significantly, ELMo and BERT have performed well for many NLP tasks, such as the GLUE benchmark, MultiNLI and the SQuAD v1.1 question answering task. However, it has also been shown that embedding performance mostly dependent on the training algorithms are chosen and on the data [21], [22]. ELMo embedding layer with BiLSTM has outperformed GloVe for emotion classification [23]. On the other hand, we see comparison of ELMo and BERT with GloVe embedding, where ELMo and BERT did not perform that well [24].

Each of the embedding types is available as both the underlying conceptual model that can be trained for specific corpora and also as pre-trained models that have been built from large training sets to be used directly or fine-tuned. In this work, we make use of pre-trained models. Specifically, we have made use of (a) GloVe pre-trained model that was trained on 2 billion tweets, 27 billion tokens, 1.2 million words and with 200 dimension vectors; (b) Word2Vec pre-trained model which includes 300 dimension word vectors for a vocabulary of 3 million words and phrases, and has been trained on 100 billion words from a Google News dataset; (c) ELMo (Small) pre-trained model that was trained on raw 1 Billion Word Benchmark [25] and has a 1024 dimension output vector; (d) BERT (Base) pre-trained model is trained on the concatenation of BooksCorpus (800M words) [26] and English Wikipedia (2,500M words) and uses 12 transformer blocks, a hidden layer of size 768 with a filter size of 3,072, and 12 self-attention heads.

B. Neural Networks

Recurrent Neural Networks (RNN) have been very popular in recent years for an understanding of the long term dependency in sentences and have shown great performance in several NLP tasks. For the same reason, contextual embedding like ELMo and BERT have utilised BiLSTM and bi-directional transformer encoders to get the context of sentences.

Several neural networks have shown great performance previously with Word2Vec, GloVe embedding models [21], [17] as they do not need explicit feature engineering, unlike many classical machine learning algorithms. Although there has been work where machine learning algorithms also showed competitive performance compared with neural networks [27], [28], [12].

However, deep networks like RNNs and CNNs might give an edge in classification by proper fine-tuning but it also tends to put too much weight on the state of the input. Considering the factor of training complexity and computational cost of different neural networks along with ELMo and BERT, this paper addresses the comparison of different embeddings with vanilla feed-forward neural network (FFNN). We chose FFNN as classifier, due to its transparency and performance for the purposes of our current feature comparison.

III. Methodology

A. Data Profile and Pre-Processing

CrisisLex [16] and CrisisNLP [6] datasets were used for this study. These datasets consist of tweets relating to earthquakes, floods, and storms from a three-year time period (2012 - 2015). The datasets consist of two levels of labelled data. Firstly, it is labelled according to the relevancy of a tweet, and then in terms of the type of information.

This paper considers 15 natural disaster datasets of various types, such as floods, earthquakes, and storms. The data is already annotated for relevant tweets. We summarise the distribution in Table I.

Table I: Distribution of Relevant Tweets

Disaster Dataset	Relevant	Not Relevant	Total
2014 California Earthquake (CalE)	2026	170	2196
2013 Pakistan Earthquake (PkE)	1677	336	2013
2014 India Floods (IF)	1501	502	2003
2014 Chile Earthquake (ChiE)	2089	364	2453
2014 Hurricane Odile Mexico (MH)	2140	56	2196
2015 Cyclone Pam (PC)	1895	718	2613
2014 Pakistan Floods (PkF)	1986	27	2013
2015 Nepal Earthquake (NE)	5792	6696	12488
2014 Philip. Typhoon Hagupit (PT)	4713	6975	11688
2013 Queensland Floods (QF)	713	179	892
2013 Typhoon Yolanda (TY)	756	168	924
2013 Colorado Floods (CF)	755	146	901
2013 Bohol Earthquake (BE)	421	522	943
2013 Alberta Floods (AF)	657	256	913
2012 Philippines Floods (PF)	744	130	874

Tweets typically consist of hashtags, URLs, punctuation, emoticons, and special characters, which would add noise to the data during training. The following workflow explains our data pre-processing:

1) Removal of hashtags, URLs, punctuation and emoticons for noise reduction.
2) Removal of data label inconsistencies:
   a) Tweets which were labelled as "Not applicable" or "Not Labeled" were removed,
   b) "Related and informative", "Relevant Information" were combined as "Informative" whereas "Related - but not informative", "Not related", and "Not Relevant" were combined as "Not_Informative",
   c) "Injured and Dead People", "Missing and Found", and "Displaced People" in CrisisNLP were combined as "Affected Individual" to make data consistent with the labelling in CrisisLex,
   d) "Money", "Volunteered services" and "Donation" were combined into the category "Donation and Volunteering",
   e) The "Other Useful Information" labelled was removed, as it is an ambiguous label,
   f) "Not Informative" tweets are removed to further classify the "Informative" tweets into the 5 types of labels as set out in Table II.
3) Removal of stop words, including the 20 least frequent words, and words with less than three characters for dimensionality reduction.
4) Lemmatization for removal of inflectional endings (returns the base or dictionary form of a word).
5) Removal of duplicate tweets using either tweet id, or text via Cosine Similarity (for those with over 90% similarity).

For second-level classification we used the following categories:

- Affected Individual (AI)
- Caution and Advice (CA)
- Donation and Volunteering (DV)
- Infrastructure and Utilities (IU)
- Sympathy and support (SS)

Table II: Information Type Distribution

Type	AI	CA	DV	IU	SS
Count	4031	1696	4645	2377	2607

B. Model

We selected a single classification architecture that could take input from the various embedding options, as evaluating the relative strength of different embeddings was our goal. For this purpose, we made use of the vanilla feed-forward neural network (FFNN) for model comparison due to the transparency of the model and its ease of explanation while providing high-performance values. We also tested against some other classification types in early training but found the feed-forward neural network gave us the best balance of transparency and performance for the purposes of our current feature comparison.

We fed each pre-trained model embedding layer to two fully connected (FC) layer of 128 units. Leaky ReLU activation function with alpha 0.1 are used after each layer to converge the model faster and overcome the dying ReLU problem that is instead of having zero slope for each $x<0$ Leaky ReLU uses small negative slope. In order to avoid over-fitting, we applied both dropout [29] of 0.5 after the each fully connected layer and also L2 regularisation of 0.001 in each layer. For the output layer, a softmax function has been used with an output of 2 or 5 dimension logit, depending on the classification task. We summarise this architecture in Figure 1.

For all models we used the Adam optimiser for training [30] with varying learning rates for embeddings according to their best performance. For BERT we used a learning rate of 5e-5 and trained for 4 epoch, while for all other embedding models learning rate used is 2e-4 and 5e-4 with epoch size 25. For the loss function, we used categorical cross-entropy. Both models use the maximum sequence length parameter to process the input sentence at once. Due to tweets' limitation to 140 characters, we kept the maximum sequence length to 128. For training of the models, we used Google Colab's GPU, which provides Tesla K80 GPU with 12 GB GDDR5 VRAM.

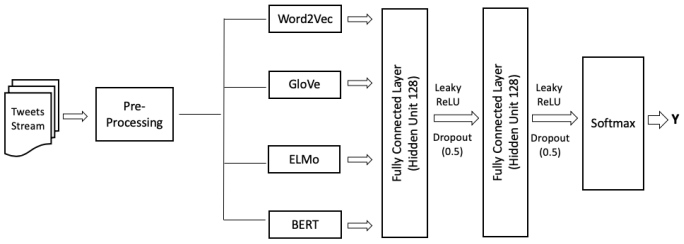

Figure 1: Model Architecture

Table III: LOO Informativeness Classification Result

Model	P	R	A	F1
Word2Vec	0.83	0.80	0.80	0.80
GloVe	0.83	0.80	0.80	0.80
ELMo	0.84	0.80	0.80	**0.81**
BERT	0.83	0.77	0.77	0.78

IV. Result

The evaluation metric used was a weighted average F1 score due to uneven class distribution.

In a Leave-one-out (LOO) approach we tested our model on one dataset individually and trained on 14 other datasets. Consequently, the model trained 15 times and we calculated the average of weighted F1 score of all test results. For informativeness classification in Table III ELMo achieved the best performance whereas for information type classification in Table IV Word2Vec showed the best result. However, the embeddings show only an insignificant difference in their performance.

Table IV: LOO Information Type Classification Result

Model	P	R	A	F1
Word2Vec	0.78	0.76	0.76	**0.76**
GloVe	0.77	0.75	0.75	0.75
ELMo	0.76	0.76	0.75	0.75
BERT	0.77	0.75	0.75	0.75

In a cross-disaster approach, we trained our model on one type of disaster data at a time and tested the model on different disaster datasets individually; we subsequently calculated the average of weighted F1 score of all test result.

Results in Table V and Table VI indicate that classification is independent of the type of disaster, as all type of disasters appears to share similar vocabularies, such as prayers, donations, injured people or needs.

For both Informativeness and Information type results, Word2Vec, ELMo, and Glove were comparable for the best F1 score, depending on the type of disaster training data.

In terms of training time, BERT and ELMo took almost 2 to 3 hours to run 4 and 25 epochs respectively for all the iterations on GPU while it takes 4-5 hours on CPU. On the other hand, Word2Vec and GloVe took 12-15 minutes on both CPU and GPU, considering the similar result of all models, Word2Vec and GloVe are also more efficient in terms of computational cost.

Table V: Cross Disaster Informativeness Classification Result

Trained on	Model	P	R	A	F1
Earthquake Data	Word2Vec	0.83	0.75	0.75	0.78
	GloVe	0.83	0.77	0.77	0.79
	ELMo	0.84	0.79	0.79	**0.81**
	BERT	0.83	0.76	0.76	0.78
Flood Data	Word2Vec	0.79	0.73	0.73	0.73
	GloVe	0.80	0.77	0.77	**0.76**
	ELMo	0.80	0.75	0.75	0.75
	BERT	0.80	0.76	0.76	0.75
Storm Data	Word2Vec	0.83	0.76	0.76	0.78
	GloVe	0.84	0.74	0.74	0.77
	ELMo	0.84	0.78	0.78	**0.80**
	BERT	0.83	0.75	0.75	0.77

Table VI: Cross Disaster Information Type Classification Result

Trained on	Model	P	R	A	F1
Earthquake Data	Word2Vec	0.74	0.70	0.70	**0.69**
	GloVe	0.72	0.69	0.69	0.67
	ELMo	0.73	0.69	0.69	0.68
	BERT	0.67	0.73	0.68	0.68
Flood Data	Word2Vec	0.76	0.72	0.72	**0.72**
	GloVe	0.73	0.69	0.69	0.68
	ELMo	0.76	0.70	0.70	0.70
	BERT	0.72	0.67	0.67	0.66
Storm Data	Word2Vec	0.74	0.71	0.71	0.71
	GloVe	0.74	0.72	0.72	**0.72**
	ELMo	0.75	0.71	0.71	0.71
	BERT	0.74	0.69	0.69	0.69

V. Discussion

The advantage of ELMo and BERT over Word2Vec and GloVe is that they capture the position of the word in the sentence. This overcomes the out of vocabulary (OOV) issue, where BERT utilises the WordPiece tokenisation embeddings [31] and ELMo uses a character-based approach. Unlike Word2Vec this allows unknown or rare words to have a representation. We anticipated that this would result in ELMo and BERT performing best. However, due to the fact that Twitter is limited to 140 character, highly informal position embeddings do not seem to give an added advantage. Additionally, in spite of resolving the OOV issue, twitter appears to contain a significant amount of unknown words as part of hashtags, which cannot be captured by pre-trained models.

BERT also trains using masking of 15% tokens and considers both previous and next-word prediction. But for the same reason as discussed in the context of ELMo it appears difficult to utilise such features of the model. These might be great with long term dependencies but might not be advantageous with short informal texts like twitter.

As all four pre-trained embedding models' performance is similar, computational cost can also be one factor to be considered. Where Word2Vec and GloVe are handy and take less time to train, BERT and ELMo are computationally costly.

Considering the high computational cost of BERT and ELMo, it is challenging to optimise the model without a good GPU or CPU, in order to improve the result.

We also analysed the wrongly predicted tweets from all models and encountered the following common issues: (i) some of the tweets were incorrectly annotated, (ii) after performing data processing some tweets are left with only one to three words, which rarely results in a meaningful message, and (iii) too many hashtags in one tweet with joint words, for example *#ineedwater, #prayforchile, #itisdestroyed*, which result in ambiguous classifications. Furthermore, it has been observed that all the worst performing test datasets had the most popular or top words as hashtags, for example, *#rescueph, #yycflood, #abflood, #bigwet*, while test datasets that performed better do not contain such hashtags.

This shows that a more fine-grained approach to pre-processing is needed in the case of tweet classification in order to improve the overall performance of the models.

VI. Conclusion

The cross-disaster result showed that disaster tweet classification is independent of the type of disaster. This aspect could be helpful in creating a robust model for the application using real-time streaming data. The results indicate that the performance of embeddings depend on the type of training data used as there is no single winner. Additionally, although BERT and ELMo achieved competitive results for most NLP tasks, their performance is quite similar to the Word2Vec and GloVe for disaster tweet classification. This could be due to the informal structure of tweets and the character limit where position embeddings do not provide an added advantage to the classification.

References

[1] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen, "Microblogging during two natural hazards events: what twitter may contribute to situational awareness," in *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2010, pp. 1079–1088.

[2] Z. Ashktorab, C. Brown, M. Nandi, and A. Culotta, "Tweedr: Mining twitter to inform disaster response." in *ISCRAM*, 2014.

[3] Q. Huang and Y. Xiao, "Geographic situational awareness: mining tweets for disaster preparedness, emergency response, impact, and recovery," *ISPRS International Journal of Geo-Information*, vol. 4, no. 3, pp. 1549–1568, 2015.

[4] B. E. Parilla-Ferrer, P. Fernandez, and J. Ballena, "Automatic classification of disaster-related tweets," in *Proc. International conference on Innovative Engineering Technologies (ICIET)*, vol. 62, 2014.

[5] M. Imran, C. Castillo, J. Lucas, P. Meier, and S. Vieweg, "Aidr: Artificial intelligence for disaster response," in *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 2014, pp. 159–162.

[6] M. Imran, P. Mitra, and C. Castillo, "Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages," may 2016. [Online]. Available: http://arxiv.org/abs/1605.05894

[7] S. Si, M. Win, and T. N. Aung, "Automated Text Annotation for Social Media Data during Natural Disasters," *Advances in Science, Technology and Engineering Systems Journal*, vol. 3, no. 2, pp. 119–127, 2018.

[8] G. Salton and M. J. McGill, "Introduction to modern information retrieval," 1986.

[9] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[10] F. Sun, J. Guo, Y. Lan, J. Xu, and X. Cheng, "Inside out: Two jointly predictive models for word representations and phrase representations," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[12] F. Alam, M. Imran, and F. Ofli, "CrisisDPS: Crisis Data Processing Services," in *In Proceedings of the 16th International Conference on Information Systems for Crisis Response and Management (ISCRAM), 2019, Valencia, Spain*, 2019. [Online]. Available: http://www.wis.ewi.tudelft.nl/twitcident/

[13] K. Rudra, P. Goyal, N. Ganguly, P. Mitra, and M. Imran, "Identifying sub-events and summarizing disaster-related information from microblogs," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 2018, pp. 265–274.

[14] F. Alam, F. Ofli, and M. Imran, "Crisismmd: Multimodal twitter datasets from natural disasters," in *Twelfth International AAAI Conference on Web and Social Media*, 2018.

[15] N. Said, K. Ahmad, M. Regular, K. Pogorelov, L. Hassan, N. Ahmad, and N. Conci, "Natural disasters detection in social media and satellite imagery: a survey," *arXiv preprint arXiv:1901.04277*, 2019.

[16] A. Olteanu, C. Castillo, F. Diaz, and S. Vieweg, "CrisisLex: A Lexicon for Collecting and Filtering Microblogged Communications in Crises," *In Proceedings of the 8th International AAAI Conference on Weblogs and Social Media (ICWSM'14)*, no. January 2014, 2014.

[17] G. Burel and H. Alani, "Crisis event extraction service (crees)-automatic detection and classification of crisis-related content on social media," 2018.

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[19] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[20] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[21] D. T. Nguyen, K. A. Al Mannai, S. Joty, H. Sajjad, M. Imran, and P. Mitra, "Robust classification of crisis-related data on social networks using convolutional neural networks," in *Eleventh International AAAI Conference on Web and Social Media*, 2017.

[22] S. E. M. O'Keefe and R. M. M. Alrashdi, "Deep learning and word embeddings for tweet classification for crisis response," in *The 3rd National Computing Colleges Conference*. York, 2018.

[23] J. A. Balazs, E. Marrese-Taylor, and Y. Matsuo, "Iiidyt at iest 2018: Implicit emotion classification with deep contextualized word representations," *arXiv preprint arXiv:1808.08672*, 2018.

[24] P. Zhong and C. Miao, "ntuer at semeval-2019 task 3: Emotion classification with word and sentence representations in rcnn," *arXiv preprint arXiv:1902.07867*, 2019.

[25] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson, "One billion word benchmark for measuring progress in statistical language modeling," *arXiv preprint arXiv:1312.3005*, 2013.

[26] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *arXiv preprint arXiv:1506.06724*, 2015.

[27] V. K. Neppalli, C. Caragea, and D. Caragea, "Deep neural networks versus naive bayes classifiers for identifying informative tweets during disasters." in *ISCRAM*, 2018.

[28] G. Burel, H. Saif, M. Fernandez, and H. Alani, "On semantics and deep learning for event detection in crisis situations," 2017.

[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[31] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.