# An Empirical Survey of Unsupervised Text Representation Methods on Twitter Data

Lili Wang[1], Chongyang Gao[2], Jason Wei[3], Weicheng Ma[4], Ruibo Liu[5], and Soroush Vosoughi[6]

[1,2,4,5,6]Department of Computer Science, Dartmouth College
[3]ProtagoLabs
[1,2,4,5]{first.last.gr}@dartmouth.edu
[3]jason@protagolabs.com
[6]soroush.vosoughi@dartmouth.edu

## Abstract

The field of NLP has seen unprecedented achievements in recent years. Most notably, with the advent of large-scale pre-trained Transformer-based language models, such as BERT, there has been a noticeable improvement in text representation. It is, however, unclear whether these improvements translate to noisy user-generated text, such as tweets. In this paper, we present an experimental survey of a wide range of well-known text representation techniques for the task of text clustering on noisy Twitter data. Our results indicate that the more advanced models do not necessarily work best on tweets and that more exploration in this area is needed.

## 1 Introduction

Recent years have witnessed an exponential increase in the usage of social media platforms. These platforms have become an important part of politics, business, entertainment, and general social life. Correspondingly, the amount of data generated by users on these platforms has also grown exponentially. Though data on social media includes various modalities, such as images, videos, and graphs, text is by far the largest type of data generated by users. Thus, in order to extract knowledge and insight from social media, sophisticated text processing models are needed. Luckily, in parallel to the growth of social media, there has been a rapid rise in the development of sophisticated text representation techniques, the most recent being large-scale pre-trained language models that use Transformer-based architecture (Vaswani et al., 2017)(such as BERT (Devlin et al., 2018), and XLNet(Yang et al., 2019)). These methods can generate general-purpose vector representations of documents that can be used for any downstream task (e.g., sentiment classification).

However, the representation power of these methods for data from social media is not well understood. This is especially true for tweets which are usually short, noisy, and idiosyncratic. This paper is an attempt to evaluate and catalogue the representation power of a wide range of methods for tweets, starting from very simple bag-of-words representations (or embeddings) to representations generated by recent Transformer-based models, such as BERT. Since we are interested in the general representation power of the methods and not their performance on any specific downstream tasks, we do not fine-tune any of the methods using downstream tasks and use unsupervised evaluation (i.e., clustering) for our survey.

## 2 Text Representation Methods

In this section, we briefly introduce the methods used in our survey, sorted from oldest to newest. For word embedding methods like word2vec, GloVe, and fastText, which dot not explicitly support sentence embeddings, we average the word embeddings to get sentence embeddings. For deep models like ELMo, BERT, ALBERT, and XLNet, we take the average of the hidden state of the last layer on the input sequence axis. Note that some other works use the hidden state of the first token ([CLS]), but in our experiments, we use the pre-trained model without fine-tuning, in this case, the hidden state of [CLS] is not a good sentence representation. Note that we use all these deep neural models without fine-tuning. This is because fine-tuning is usually based on specific downstream tasks which bias the information in the hidden states, weakening the general representation. Note that when we refer to n-gram models we mean models that capture all grams up to and including the n-gram (e.g., bigram models will include bigrams and unigrams).



*Proceedings of the 2020 EMNLP Workshop W-NUT: The Sixth Workshop on Noisy User-generated Text*, pages 209–214
Online, Nov 19, 2020. 

**1. bag-of-words (BoW).** This is a representation of text that describes the occurrence of words within a document. In our experiments, we use a random sample of 5 million tweets collected from the Internet Archive Twitter dataset [1] (IAT) to create a vocabulary. We also remove stop words from the tweets. We try unigram, bigram, and trigram models.

**2. TF-IDF.**. Term frequency–inverse document frequency (TF-IDF) reflects how important a word is with respect to documents in a collection or corpus. We use a similar experimental setup as BoW.

**3. LDA** (Hoffman et al., 2010). Latent Dirichlet allocation (LDA) is a generative statistical model for capturing the topic distribution of documents in a corpus. We train this model on the IAT dataset. We also remove stop-words and train models with 5, 10, 20, and 100 topics.

**4. word2vec** (Mikolov et al., 2013). word2vec is a distributed representation of words based on a model trained on predicting the current word from surrounding context words (CBOW). We train unigram, bigram, and trigram word2vec models using the IAT dataset.

**5. doc2vec** (Le and Mikolov, 2014). This model extends word2vec by adding another document vector based on ID. Our model is trained on the IAT dataset.

**6. GloVe** (Pennington et al., 2014). This model combines global matrix factorization and local context window methods for training distributed representations. We use the 200-dimensional version that was pre-trained on 2 billion tweets.

**7. fastText** (Joulin et al., 2016). fastText is another word embedding method that extends word2vec by representing each word as an n-gram of characters. We use the 300-dimensional off-the-shelf version which was pre-trained on Wikipedia.

**8. Tweet2vec** (Dhingra et al., 2016). This model finds vector-space representations of whole tweets by learning complex, non-local dependencies in character sequences. In our experiments, we use the pre-trained best model provided by the authors.[2]

[1] https://archive.org/search.php?query=collection%3Atwitterstream&sort=-publicdate

[2] https://github.com/bdhingra/tweet2vec/tree/master/tweet2vec/best_model There is another tweet2vec model that uses a character-level cnn-lstm encoder-decoder (Vosoughi et al., 2016), but for the sake of brevity we only show the results for one of the tweet2vec models.

**9. Universal Sentence Encoder (USE)** (Cer et al., 2018). USE encodes sentences into high dimensional vectors. The pre-trained encoder comes in two versions, one trained with deep averaging network (DAN) (Iyyer et al., 2015) and one with Transformer. We use the DAN version of USE.

**10. ELMo** (Peters et al., 2018). This method provides context-dependent word representations based on bidirectional language models. We use the version pre-trained on the One Billion Word Benchmark.

**11. BERT** (Devlin et al., 2018). BERT is a large-scale Transformer-based language representation model (Vaswani et al., 2017). We use two off-the-shelf pre-trained versions BERT-base and BERT-large, which are pre-trained on the BooksCorpus and English Wikipedia respectively.

**12. ALBERT** (Lan et al., 2019). This is a lite version of BERT, with far fewer parameters. We use two off-the-shelf versions, ALBERT-base and ALBERT-large, which are pre-trained on the BooksCorpus and English Wikipedia respectively.

**13. XLNet** (Yang et al., 2019). This is an autoregressive Transformer-based language model. Like BERT, XLNet is a large-scale language model with millions of parameters. We use the off-the-shelf versions pre-trained on the BooksCorpus and English Wikipedia.

**14. Sentence-BERT** (Reimers and Gurevych, 2019). Sentence-BERT modifies BERT by using siamese and triplet network structures to derive semantically meaningful sentence embeddings. We use five off-the-shelf versions provided by the authors, Sentence-BERT-base, Sentence-BERT-large, Sentence-Distilbert, Sentence-RoBERTa-base, and Sentence-RoBERTa-large, all pre-trained on NLI data.

## 3 Experiments

Since we are interested in measuring the general text representation power of our methods, we use clustering as a way to evaluate the representations generated by each model (instead of any downstream supervised tasks). We use the vector representations of each tweet to run $k$-means clustering for different values of $k$. We use two tweet datasets for our evaluation. The tweets in these datasets have labels corresponding to their topic which we use as cluster ground-truth for evaluation purposes.

**Dataset 1** (Zubiaga et al., 2015): This dataset includes 356,782 tweets belonging to 1,036 topics.

We use $k \in \{200, 400, 600, 800, 1000\}$, for this dataset.

**Dataset 2** (Rosenthal et al., 2017): This dataset includes 35,323 tweets belonging to 374 topics. We use $k \in \{100, 200, 300, 400, 500\}$, for this dataset.

### 3.1 Evaluation Metrics

We use a total of six metrics for evaluating the "goodness" of our clusters, described below. Except for the Silhouette score, all other metrics rely on ground-truth labels.

**Silhouette score** (Rousseeuw, 1987): A good clustering will produce clusters where the elements inside the same cluster are close to each other and the elements in different clusters are far from each other. The Silhouette score takes both these factors into account. The score goes from -1.0 to 1.0, where higher values mean better clustering.

**Homogeneity, Completeness, and V-measure**, (Rosenberg and Hirschberg, 2007): If clusters contain only data points that are members of a single class, in other words, high homogeneity, this usually indicates good clustering. Similarly, if all members of a given class are assigned to the same cluster, in other words, high completeness, this usually indicates good clustering. The Homogeneity and Completeness scores are between 0.0 and 1.0, where higher values correspond to better clustering. The V-measure score is the harmonic mean of Homogeneity and Completeness.

**Adjusted Rand Index (ARI)** (Hubert and Arabie, 1985): The Rand Index can be used to compute the similarity between generated clusters and ground-truth labels. This is done by considering all pairs of samples and seeing whether their label agreement (i.e., belonging to the same ground-truth cluster or not) matches the generated cluster agreement (i.e., belonging to the same generated cluster or not). The raw RI score is then "adjusted for chance" into the ARI. score using the following formula: The ARI score can be between -1.0 and 1.0, where random clusterings have an ARI close to 0.0 and 1.0 stands for perfect clustering.

**Adjusted Mutual Information (AMI)** (Vinh et al., 2010): The Mutual Information (MI) score is an information-theoretic metric that measures the amount of "shared information" between two clusterings. The Adjusted Mutual Information (AMI) is an adjustment of the Mutual Information (MI) score to account for chance. It accounts for the fact that the MI is generally higher for two clusterings with a larger number of clusters, regardless of whether there is actually more information shared. The AMI score can be between 0.0 and 1.0, where random clusterings have an AMI close to 0.0 and 1.0 stands for perfect clustering.

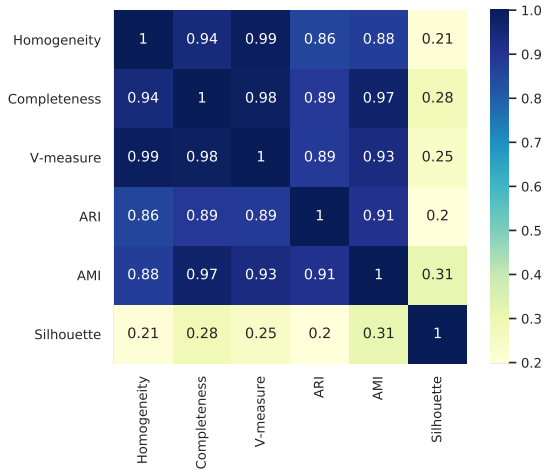


Figure 1: Confusion matrix of the correlation (Pearson's r) between each pair of methods.

## 4 Results & Discussion

For each dataset, we average the scores from $k$-means clustering with different values of $k$. Though we use several metrics in our evaluations for the sake of being thorough, most of the metrics are in fact highly correlated. Fig. 1 shows the correlation between each pair of metrics (calculated based on the clustering results of our methods). We can see that all the *external* evaluation metrics (Homogeneity, Completeness, V-measure, AMI, and ARI, which need external ground-truth labels) highly agree with each other while the *internal* evaluation metric (Silhouette score, which does not need external ground-truth labels) does not.

The clustering results are shown in Fig. 2 and Fig. 3, the methods in both figures are sorted based on the date of their release to capture the advancements in NLP. Unlike conventional tasks and datasets (such as the GLUE benchmark (Wang et al., 2018)), there does not seem to be a very clear trend of improvement for capturing tweet representations. The more advanced models are not necessarily the best. Notably, the BERT family of large-scale pre-trained language models (ALBERT, Sentence-BERT, etc) do not vastly or consistently outperform much simpler methods such as bag-of-words and tf-idf. XLNet, on the other hand, seems to be the best performing method for cap-

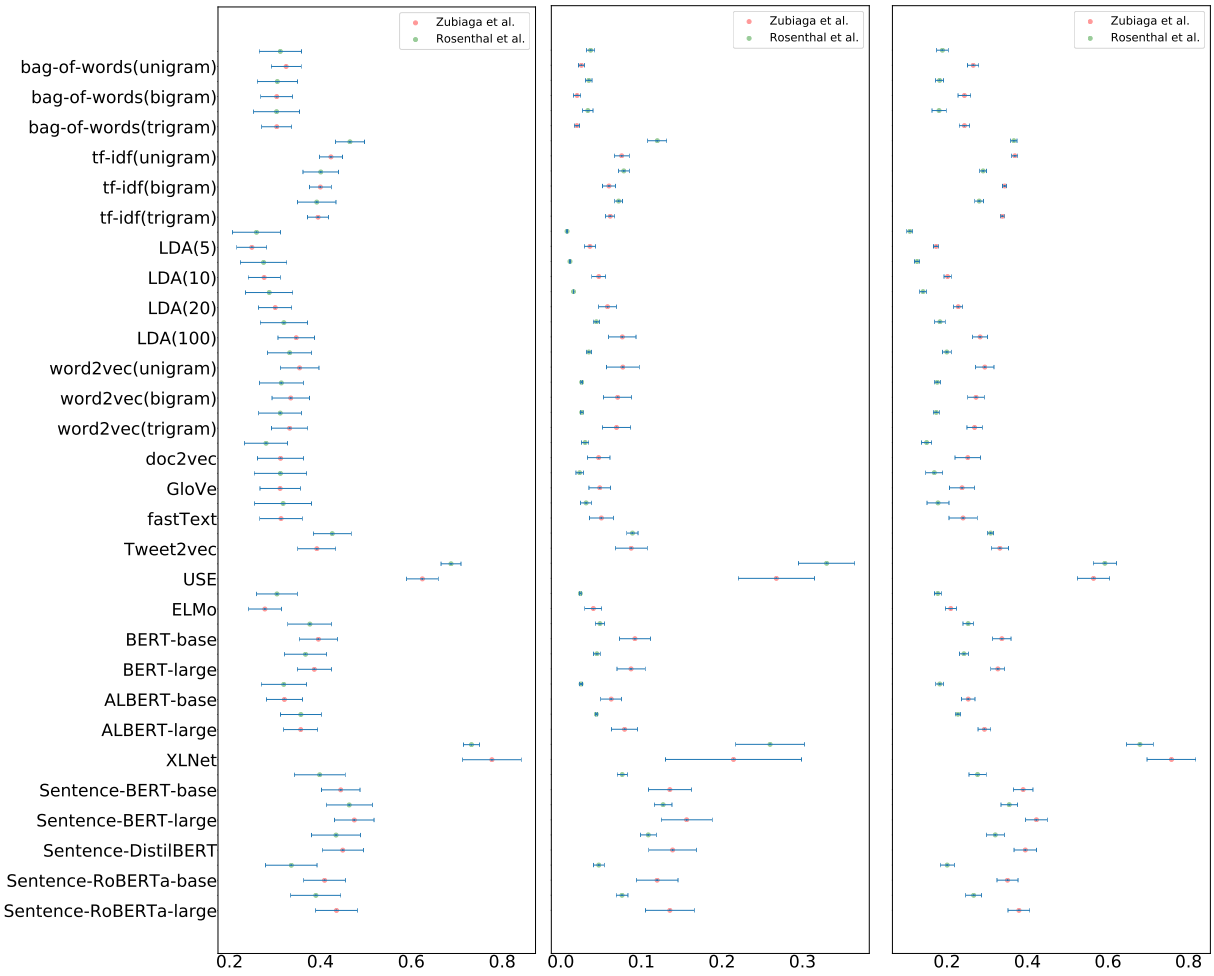


Figure 2: The V-measure (left), ARI (middle), and AMI (right) of all the methods on the two datasets. The points in the figure denote the average value across different $k$ values and the blue lines denote the standard deviations. The methods are sorted from the oldest to the newest.

turing tweet representations, followed closely by USE. Interestingly, XLNet is also the most volatile with respect to the choice of $k$ in our clustering. We think XLNet outperforms other comparable (in terms of complexity) models such as BERT since it uses permutation language modeling, allowing for prediction of tokens in random order. This might make it more robust to the noisy user-generated text, such as tweets. We think that our results are unexpected and inconclusive, demonstrating that much is still unknown about the performance of the most recent models on noisy and idiosyncratic user-generated text.

Very recently, a large-scale pre-trained BERT model for English Tweets was trained and released (Nguyen et al., 2020). This model was released just days before the publication of this paper and thus we did not have time to thoroughly compare its performance against the other models. However, we believe this model is a step in the right direction as we have shown in this paper that models trained on standard English corpora do not perform well on Tweets.

## 5 Conclusion

In this paper, we presented an experimental survey of 14 methods for representing noisy user-generated text prevalent in tweets. These methods ranged from very simple bag-of-words representations to complex pre-trained language models with millions of parameters. Through clustering experiments, we showed that the advances in NLP do not necessarily translate to better representation of tweet data.

We believe more work is needed to better understand and potentially improve the performance of the more recent methods, such as BERT, on noisy, user-generated data.

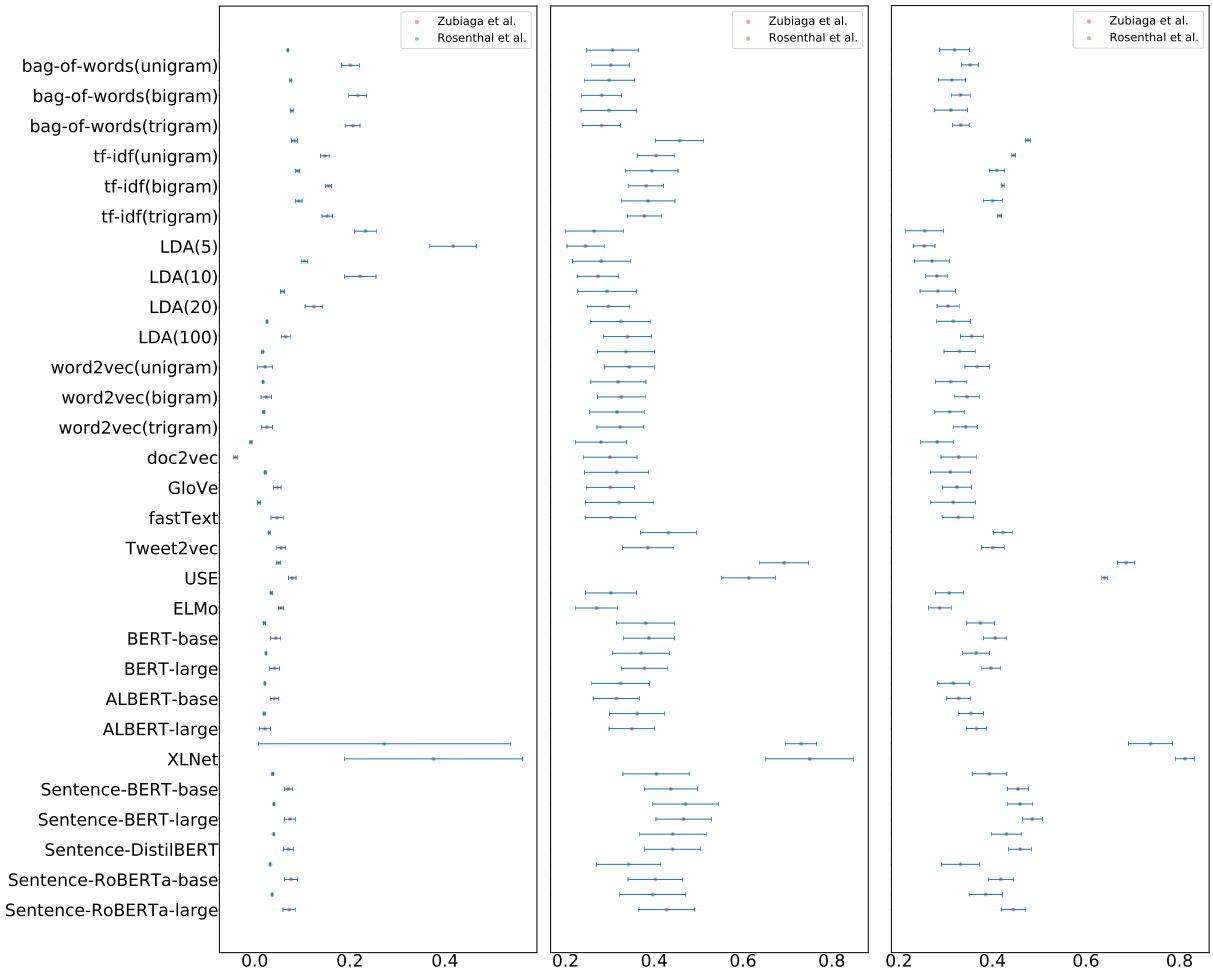


Figure 3: The Silhouette (left), Homogeneity (middle), and Completeness (right) of all the methods on the two datasets. The points in the figure denote the average value across different $k$ values and the blue lines denote the standard deviations. The methods are sorted from the oldest to the newest.

## References


Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

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

Bhuwan Dhingra, Zhong Zhou, Dylan Fitzpatrick, Michael Muehl, and William Cohen. 2016. Tweet2vec: Character-based distributed representations for social media. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 269–274, Berlin, Germany. Association for Computational Linguistics.

Matthew Hoffman, Francis R Bach, and David M Blei. 2010. Online learning for latent dirichlet allocation. In *advances in neural information processing systems*, pages 856–864.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification*, 2(1):193–218.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, pages 1681–1691.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Interna-*

*tional conference on machine learning*, pages 1188–1196.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. Bertweet: A pre-trained language model for english tweets. *arXiv preprint arXiv:2005.10200*.

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

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 410–420.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518, Vancouver, Canada. Association for Computational Linguistics.

Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

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

Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2010. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research*, 11:2837–2854.

Soroush Vosoughi, Prashanth Vijayaraghavan, and Deb Roy. 2016. Tweet2vec: Learning tweet embeddings using character-level cnn-lstm encoder-decoder. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 1041–1044.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Arkaitz Zubiaga, Damiano Spina, Raquel Martínez, and Víctor Fresno. 2015. Real-time classification of twitter trends. *Journal of the Association for Information Science and Technology*, 66(3):462–473.