Semantic Textual Similarity of Sentences with Emojis

Alok Debnath
Kohli Center for Intelligent Systems
International Institute of Information Technology, Hyderabad
alok.debnath@research.iiit.ac.in

Nikhil Pinnaparaju
Kohli Center for Intelligent Systems
International Institute of Information Technology, Hyderabad
nikhil.pinnaparaju@research.iiit.ac.in

Manish Shrivastava
Kohli Center for Intelligent Systems
International Institute of Information Technology, Hyderabad
m.shrivastava@iiit.ac.in

Vasudeva Varma
Kohli Center for Intelligent Systems
International Institute of Information Technology, Hyderabad
vv@iiit.ac.in

Isabelle Augenstein
Department of Computer Science
University of Copenhagen
augenstein@di.ku.dk

ABSTRACT

In this paper, we extend the task of semantic textual similarity to include sentences which contain emojis. Emojis are ubiquitous on social media today, but are often removed in the pre-processing stage of curating datasets for NLP tasks. In this paper, we qualitatively ascertain the amount of semantic information lost by discounting emojis, as well as show a mechanism of accounting for emojis in a semantic task. We create a sentence similarity dataset of 4000 pairs of tweets with emojis, which have been annotated for relatedness. The corpus contains tweets curated based on common topic as well as by replacement of emojis. The latter was done to analyze the difference in semantics associated with different emojis. We aim to provide an understanding of the information lost by removing emojis by providing a qualitative analysis of the dataset. We also aim to present a method of using both emojis and words for downstream NLP tasks beyond sentiment analysis.

KEYWORDS

datasets, emoji, sentence similarity

ACM Reference Format:
Alok Debnath, Nikhil Pinnaparaju, Manish Shrivastava, Vasudeva Varma, and Isabelle Augenstein. 2020. Semantic Textual Similarity of Sentences with Emojis. In *Companion Proceedings of the Web Conference 2020 (WWW '20 Companion), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3366424.3383758

1 INTRODUCTION

Social media is a goldmine of raw data for semantic processing tasks such as sarcasm and humour detection, sentence similarity and entity or event relations. However, social media data is user generated text, which is highly noisy and sparse. Therefore, data mined from social media requires preprocessing for removing noise, which results in loss in information [18].

More often than not, semantic classification tasks treat emojis as noise and remove them from the dataset in the pre-processing stage [11]. However, due to their ubiquity and variety, emojis contain semantic information. The work that exists on taking emojis into account, for sentiment analysis and sarcasm detection, demonstrates that utilising the semantic information they carry is beneficial [7, 17]. With work in emoji embeddings and representation in vector spaces [3, 6], as well as some work in their semantic analysis and comparison [19], we find that emojis can be represented, processed and compared as semantic units. Therefore, the role played by emojis in downstream NLP tasks and their associated semantics must be investigated.

In this paper, we propose to analyse this phenomenon in more depth by studying the relationship between textual similarity and emojis. We construct a dataset of 4000 tweet pairs, and annotate them for relatedness in a manner similar to the SICK relatedness annotation [12]. We show the development of this dataset from an initial 300,000 tweets, as well as the annotation procedure. We analyze the dataset in order to provide an insight into how the similarity of sentences changes based on the emojis used. Finally, we compare the performance of common sentence similarity models on our dataset using just word embeddings as well as word and emoji embeddings, and provide a comprehensive analysis of the results of the experiments.

2 RELATED WORK

In this section, related works and recent some important developments in the NLP with emojis is highlighted, as well as current progress in sentence similarity with a focus on distributional models.

Research on the interpretation and prediction of emojis has developed in a similar spirit to other research in a NLP, with similar representation learning based methods. Advances in NLP of Emojis include affirmation of their predictability [2] and distributional representations such as emoji2vec [6], to name a few.

Barbieri et al. [3] explore a vector skip-gram model for emojis in tweets. The skip-gram model, introduced by Mikolov et al. [13], was at the time the most widely used word representation learning method, distributed as part of the word2vec package. The approach taken by Barbieri et al. [3] is based on the similarity of emojis to tokens. Eisner et al. [6] established pre-trained emoji embeddings,

---

WWW '20 Companion, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7024-0/20/04.
https://doi.org/10.1145/3366424.3383758

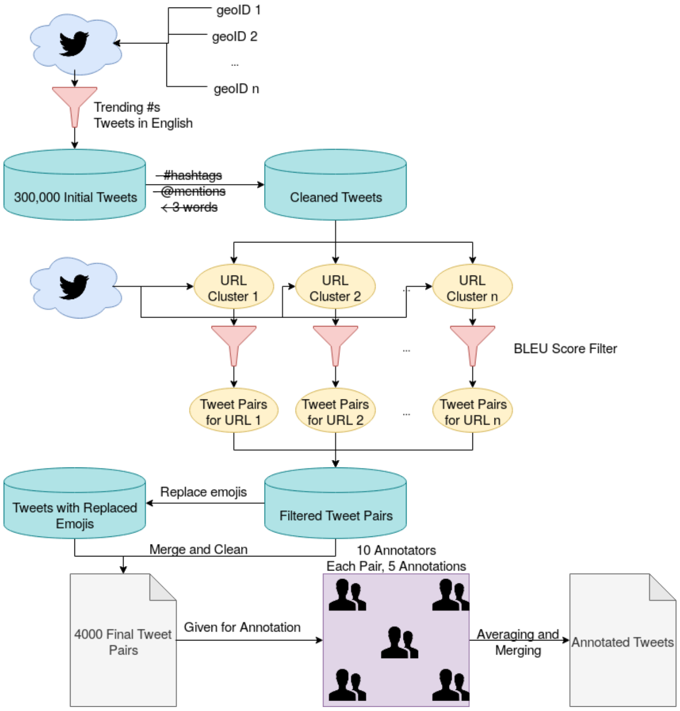
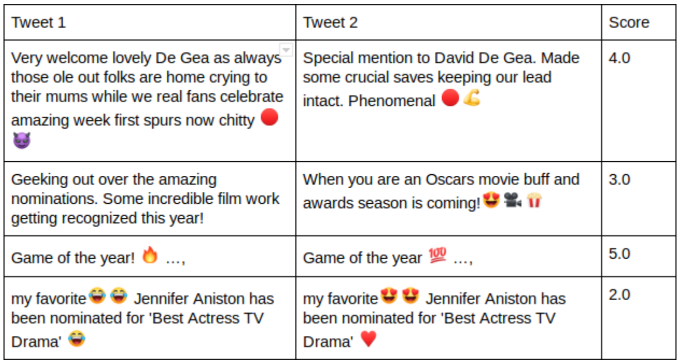


Figure 1: Data collection and annotation procedure



Tweet 1	Tweet 2	Score
Very welcome lovely De Gea as always those ole out folks are home crying to their mums while we real fans celebrate amazing week first spurs now chitty 🔴 👿	Special mention to David De Gea. Made some crucial saves keeping our lead intact. Phenomenal 🔴 💪	4.0
Geeking out over the amazing nominations. Some incredible film work getting recognized this year!	When you are an Oscars movie buff and awards season is coming!😍 🎥 🍿	3.0
Game of the year! 🔥 ...,	Game of the year 💯 ...,	5.0
my favorite😂 😂 Jennifer Aniston has been nominated for 'Best Actress TV Drama' 😂	my favorite😍 😍 Jennifer Aniston has been nominated for 'Best Actress TV Drama' ❤️	2.0

Figure 2: Examples of curated and annotated tweets. The first two examples are directly collected from Twitter, the second two are constructed by augmentation and replacement of emojis.

aptly named emoji2vec, which when combined with word2vec, can be used easily for other NLP tasks.

Work on semantic sentence similarity includes statistical models, which enforce semantic similarity in terms of a weighted average of the occurrence of the words in a document or corpus [1], or in methods such as relatedness and entailment, which are focused on topical similarity. An exhaustive survey of sentence similarity measures [8] shows that sentence similarity is comprised of three layers of similarity, which are lexical, syntactic and semantic.

Neural approachers to sentence similarity was proposed by He et al. [9], using CNNs to capture semantic similarity between sentences, which had a Siamese structure. A similar model was emplyed in Siamese Architecture on LSTMs for sentence similarity [14]. One of the most well-known datasets for semantic sentence similarity is the SICK dataset [12].

3 DATASET DEVELOPMENT

In this section, we look into the development of the dataset for tweet similarity for tweets which contain emojis. Figure 1 shows the dataset creation and annotation procedure graphically. Figure 2 shows examples of the annotated and curated dataset, which has been made available publicly.[1]

3.1 Data Collection

We use the Twitter API[2] to first collect a list of trending topics and hashtags by geoIDs.[3] Trending topics are used to collect tweets, as we hypothesise that tweets on the same topic are more likely to have a high semantic similarity. The language of tweets is restricted to English, to avoid issues arising from code mixing and code switching, which we consider outside the scope of this paper.

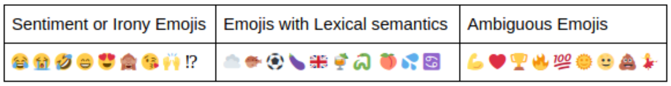
A preliminary corpus in the order of 300,000 tweets was collected. These tweets were cleansed by removing hashtags and mentions, and were filtered based on sentence length and the number of emojis. Sentences with fewer that three words were removed entirely. The remaining corpus was then organized in pairs based on the `URL`s present in them, i.e. tweets with the same URL were clustered together and then divided into pairs. Tweets with multiple `URL`s were placed in multiple clusters. The intuition behind creating pairs based on URLs was that they would be good candidates for semantically similar tweet pairs.

For each `URL`, we search Twitter again for English language tweets and added new tweets to the `URL` cluster. Each tweet in a cluster is cleansed by removing `URL`s as well. We compute the BLEU score [15] for each pair of tweets and remove those which have too low or too high a BLEU score, as this can be seen as noise and might skew the dataset. Approximately 40,000 tweets were removed from the original set due to too low BLEU score and another 44,000 were removed because their BLEU score values were too high. Repeated pairs are removed and the clusters are then merged and shuffled.

We also augment this dataset by modifying the emojis used in the tweets. For each tweet that contains an emoji, we replace it with one of the top 10 most popular emojis[4]. These constructed tweets are then paired with the original tweets. These pairs are added in order to study how the semantic information represented by emojis in a context changes the meaning of tweets. These constructed tweets are then added to the dataset. It was found that the most common emojis were usually associated with sentiment and/or irony. This has been detailed in 3.3.

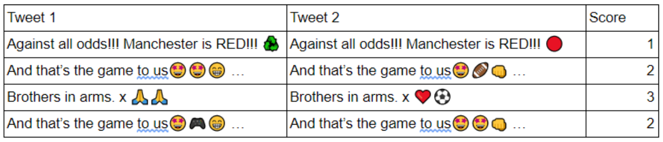
---

[1] https://drive.google.com/drive/folders/11KqRu4VYX4J7VDcLDUQkESrL3N3GO7Eb
[2] https://developer.twitter.com/en/docs/api-reference-index
[3] http://woeid.rosselliot.co.nz/
[4] Curated from the http://emojitracker.com/ API



Sentiment or Irony Emojis	Emojis with Lexical semantics	Ambiguous Emojis
😂😭🤣😄😍🙈😘🙌⁉	☁🦔⚽🍆🇬🇧🍹➰🍑💦♋	💪❤🏆🔥💯🌞🙂💩💃

Table 1: Examples of emojis that denote sentiment and irony, as well as semantically ambiguous emojis



Tweet 1	Tweet 2	Score
Against all odds!!! Manchester is RED!!! ♻	Against all odds!!! Manchester is RED!!! 🔴	1
And that's the game to us🤩🤩😁 …	And that's the game to us🤩🏈👊 …	2
Brothers in arms. x 🙏🙏	Brothers in arms. x ❤⚽	3
And that's the game to us🤩🎮😁 …	And that's the game to us🤩🤩👊 …	2

Table 2: Examples of semantic shift due to different emojis being used

3.2 Annotation Procedure

Once the dataset was collected, we created annotation guidelines for labelling the semantic similarity for each tweet pair. The guidelines used are a modification of the SICK annotation guidelines for measuring relatedness [12]. Annotators were asked to mark how similar they perceived two tweets to be, in terms of their how close their meanings are. Annotators were asked to provide higher score to those tweet pairs which are partly synonymous with one another, disregarding additional semantic content in one over the other, provided the semantic content does not change the tone of the tweets.

We use ten annotators for this project between the ages of 17 and 22, who are well versed with Twitter and the meanings of emojis in context. The corpus is split into half and the annotators are provided one of the two sets to label. Therefore, each pair of tweets is annotated five times. Each annotator labels each tweet pair with a score from 1 to 5, 1 being very dissimilar and unrelated tweets, and 5 being the identical or very similar tweets. Each annotator was required to consider the contribution of the emojis in the semantic content of the tweet itself, but was not told the nature of the task at hand so as to reduce the annotators' bias towards the task. After having collated the annotations, scores are averaged across annotators.

3.3 Analyzing the Dataset

In this section, we provide a detailed analysis of the dataset. We find that in emojis highly correlated with sentiment and irony provide little semantic content, and are also the most popular emojis. However, if an emoji associated with sentiment or irony was used in place of an emoji with lexical semantic characteristics, the difference in meaning was found to be significant by almost all the annotators. There exists a class of emojis which has been used ambiguously, to provide both semantic meaning as well as provide sentiment. We find that these emojis are quite commonly used with emojis showing only sentiment, and that replacing these makes the tweets almost unrelated to each other.

We find that relatedness is directly associated with the category of the emoji being replaced. As mentioned earlier, the most common emojis are those which are associated with sentiment and/or irony.

As seen in Table 2, we find that the class of the emoji being replaced with a sentiment emoji directly affect the relatedness score. Tweets with the original and modified tweets both containing sentiment emojis show high relatedness scores (shown in example 3 on Table 2). However, emojis which have semantic information (see Example 1 and Example 4 in Table 2), upon replacement, show much lower relatedness scores. We attribute this to the stripping of semantic information when replaced with a sentiment emoji, akin to the replacement of a word with a smiley (or the like). Interestingly, ambiguous emojis show a spectrum of relatedness scores on replacement. This is because they carry both sentiment and semantic information. When replaced with emojis of a similar sentiment, the emojis show high (but not complete) relatedness. However, if the sentiment of the emoji is vastly different from that of the original in the text, there is a large drop in the relatedness score (again seen in Example 1).

4 EXPERIMENTS

In this section, we experiment with different models and baseline metrics for sentence similarity. To show the difference between accounting for emojis in the task of sentence similarity or not on our dataset, we use word embeddings both with and without emoji representations. Given that sentence similarity is typically framed as a regression task, which we also follow here, we use *mean squared error* as our loss function, along with the Pearson Correlation Metric [5]. Our experiments use the *Adadelta* optimiser [20] for all of our models.

4.1 Models and Embeddings

For experiments on sentence similarity, we use three models:

1. **LSTM + FC** - An LSTM Sentence Encoder [10] applied to the individual sentences. The final outputs for each sentence of the sentence pair is taken and concatenated with the other and passed to a fully connected layer reducing the dimensionality to 1. This model serves as a baseline model for this task and provides intuition about the relative performance of more complex neural models for semantic similarity.
2. **MaLSTM** - The Manhattan LSTM Neural Network architecture (MaLSTM) is a popular architecture that uses the Manhattan distance to compute the difference between two input sentences, and so it can be used for sentence similarity tasks [14]. MaLSTM is the state-of-the-art in semantic sentence similarity tasks and therefore can be used to determine how well this model does with and without emoji information.
3. **CASNN** - Cross Attention Siamese Neural Network, or CASNN, is a Siamese network that employs cross attention on individual embeddings to generate a feature representation of two input sentences from a bidirectional LSTM, which is then normalized and compared for computing similarity [21]. The CASNN model was used to motivate the use of attention based architectures for similarity tasks which provide relative importance at a lexical (or emoji) level.

Each model is given four different input embeddings, which is done in order to compare the performance of pretrained embeddings both with and without emojis. We use word2vec [13] and GloVe

[16] both, in order to showcase the difference associated with using global vector representation pretrained on Twitter data, as opposed to the vector space being shared by the embeddings with emojis. For embeddings with emojis, we use word2vec with emoji2vec [6] and the combined skip-gram model Barbieri et al. [4]. The word2vec and emoji2vec embeddings have words and emojis residing in two different vector spaces, and the combined skip-gram model provides words and emojis that are present in the same vector space. We use both in order to contrast the performance of using two independent vector spaces for words and emojis versus the same vector space as Mikolov et al. [13].

For each of the experiments, we perform model ablations on dimension sizes of 50, 100 and 200. The models are run on an 80-20 train-test split. In case of considering naive word2vec and GloVe, the emojis were ignored entirely, whereas for the combined models, the emoji representation was provided along with the word representation.

4.2 Results and Analysis

We present the results of the experiments described above. We also provide some analysis and insights into using these networks for the sentence similarity task and on the need to analyse emojis in NLP more widely.

Generally, GloVe embeddings perform better than the vanilla word2vec embeddings. However, performance of the emoji embeddings in conjunction with word embeddings changes with the network used for this task. However, across networks, we can see that using emoji embeddings tends to result in a lower mean squared error and higher Pearson Correlation, which can be attributed to the semantics associated with accounting for emojis. Pearson Correlation and MSE do not agree on a few of the models, as Pearson's depends of normalized covariance rather than just the average error value.

Interestingly, for sentence similarity, considering words and emojis in equivalent but different spaces improves performance as opposed to using the same space for their representation.

4.2.1 LSTM + Fully Connected Layer. First, we analyze the results of the baseline model of a naive LSTM encoder and a fully connected layer. We see here that using word2vec + emoji2vec combined outperfoms all other embeddings for this model. The combined skip-gram model does not perform well when using this simple model.

On average, increasing the hidden dimensions improves performance, but there is risk of rapid overfitting with increasing the number of hidden layers on such a simple model. We see that in the difference of trends between MSE and Pearson's scores. The word2vec + emoji2vec embeddings on 100 and 200 hidden dimensions are the best performing. GloVe shows low Mean Squared Error, but also shows low Pearson's scores. Table 3 shows the results of this model.

Hidden Dimension Size	**Word Embedding**	***MSE***	**Pearson Coefficient**
50	word2vec	1.6139	16.1149
100	word2vec	1.6147	13.6969
200	word2vec	1.6144	12.8177
50	GloVe	1.6142	9.4756
100	GloVe	1.6143	13.4933
200	GloVe	1.6144	18.7515
50	word2vec + emoji2vec	1.6146	7.6368
100	word2vec + emoji2vec	1.6143	**23.2121**
200	word2vec + emoji2vec	**1.6131**	22.2103
50	Combined Skip-gram	1.6721	9.5949
100	Combined Skip-gram	1.6711	3.8725
200	Combined Skip-gram	1.6752	11.9634

Table 3: Results of LSTM + FC on various Hidden Dimension Sizes and Embeddings

4.2.2 MaLSTM Model. The Manhattan LSTM or MaLSTM model shows higher mean squared errors. However, it also shows the highest Pearson Correlation among all the models. Naive word2vec and GloVe embeddings do not capture enough information, as is seen in the high error rates and low Pearson Coefficients. Table 4 shows the scores across dimensions sizes and embeddings.

Hidden Dimension Size	**Word Embedding**	***MSE***	**Pearson Coefficient**
50	word2vec	3.4276	26.2816
100	word2vec	3.4241	26.3179
200	word2vec	3.4228	27.1239
50	GloVe	3.4882	26.5956
100	GloVe	3.5021	25.9201
200	GloVe	3.5424	26.2441
50	word2vec + emoji2vec	3.4375	32.9631
100	word2vec + emoji2vec	**3.4209**	33.3847
200	word2vec + emoji2vec	3.4361	33.0706
50	Combined Skip-gram	3.5250	35.8767
100	Combined Skip-gram	3.5400	35.8165
200	Combined Skip-gram	3.5204	**35.9449**

Table 4: Results of Manhattan LSTM on various Hidden Dimension Sizes and Embeddings

Interestingly, while the word2vec + emoji2vec embeddings provide lower mean squared errors with the lowest at 100 hidden dimensions, we see that the combined skip-gram model provides much higher Pearson's Correlation. We conjecture that this is due to lower normalized covariance of the Manhattan distance prediction and the actual value is lower when using a single feature space for a large number of predictions, which is not possible with shared feature spaces as seen for emoji2vec+word2vec.

Hidden Dimension Size	**Word Embedding**	***MSE***	**Pearson Coefficient**
50	word2vec	1.5824	27.0829
100	word2vec	1.5822	26.1527
200	word2vec	1.5858	20.2780
50	GloVe	1.5835	22.2103
100	GloVe	1.5838	26.1345
200	GloVe	1.5852	25.8422
50	word2vec + emoji2vec	1.5822	25.9479
100	word2vec + emoji2vec	1.5823	**28.3717**
200	word2vec + emoji2vec	1.5824	9.0729
50	Combined Skip-gram	**1.5660**	19.5915
100	Combined Skip-gram	1.5662	19.9033
200	Combined Skip-gram	1.5725	18.1376

Table 5: Results of CASNN on various Hidden Dimension Sizes and Embeddings

4.2.3 *Cross Attention Siamese Bi-LSTM Model.*

The Cross Attention Siamese Neural Network Model (CASNN) employs a bidirectional LSTM with shared weights to encode the sentence, from which we calculate the relative importance of each input based on a cross attention score. We use a dropout of 0.5, i.e. drop the weights randomly in order to reduce the chances of overfitting based on the probability distributions of the generated attention scores. Table 5 shows the results of the CASNN model for various dimension sizes and embeddings.

Here too, we observe that the use of emoji information accounts for a lower MSE and higher Pearson Correlation. Interestingly, the latter is much higher for combined skip-gram, that for the other models, with comparable MSE values, which might indicate some utility in a single semantic representation of emojis and words rather than two aligned spaces. However, more experiments need to be run before this can be concluded.

5 CONCLUSION

In this paper, we present the creation of a dataset for sentence similarity for sentences with emojis. We do so in order to showcase the need to account for the processing of emojis in NLP on social media data. We highlight the development of the dataset, including cleansing, preprocessing and annotation. We run multiple experiments and model ablations on the dataset and show that accounting for emojis in a semantically driven task such as sentence/tweet relatedness provides important semantic information.

We hope to use these preliminary experiments to showcase that emojis can be used to extract more semantic content such as sarcasm, emphasis and subject matter. In the future, experiments on embedding alignment between word or character representations as well as evaluation of sentence similarity based on weighted distribution of attention can be considered on this dataset to improve results. Furthermore, contextual representations of emojis with text can prove useful for applications of NLP in social media, which can be tested on our dataset.

REFERENCES

[1] Palakorn Achananuparp, Xiaohua Hu, and Xiajiong Shen. 2008. The evaluation of sentence similarity measures. In *International Conference on data warehousing and knowledge discovery*. Springer, 305–316.

[2] Francesco Barbieri, Miguel Ballesteros, and Horacio Saggion. 2017. Are Emojis Predictable?. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, Vol. 2. 105–111.

[3] Francesco Barbieri, Francesco Ronzano, and Horacio Saggion. 2016. What does this emoji mean? a vector space skip-gram model for twitter emojis. In *Calzolari N, Choukri K, Declerck T, et al, editors. Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016); 2016 May 23-28; Portorož, Slovenia. Paris: European Language Resources Association (ELRA); 2016. p. 3967-72.* ELRA (European Language Resources Association).

[4] Francesco Barbieri, Francesco Ronzano, and Horacio Saggion. 2016. What does this Emoji Mean? A Vector Space Skip-Gram Model for Twitter Emojis. In *Language Resources and Evaluation conference, LREC*. Portoroz, Slovenia.

[5] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. Pearson correlation coefficient. In *Noise reduction in speech processing*. Springer, 1–4.

[6] Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bosnjak, and Sebastian Riedel. 2016. emoji2vec: Learning Emoji Representations from their Description. In *Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media*. 48–54.

[7] Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *arXiv preprint arXiv:1708.00524* (2017).

[8] Rafael Ferreira, Rafael Dueire Lins, Steven J Simske, Fred Freitas, and Marcelo Riss. 2016. Assessing sentence similarity through lexical, syntactic and semantic analysis. *Computer Speech & Language* 39 (2016), 1–28.

[9] Hua He, Kevin Gimpel, and Jimmy Lin. 2015. Multi-perspective sentence similarity modeling with convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 1576–1586.

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

[11] CS Pavan Kumar and LD Dhinesh Babu. 2019. Novel Text Preprocessing Framework for Sentiment Analysis. In *Smart Intelligent Computing and Applications*. Springer, 309–317.

[12] Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK Cure for the Evaluation of Compositional Distributional Semantic Models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)* (26-31), Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis (Eds.). European Language Resources Association (ELRA), Reykjavik, Iceland.

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

[14] Jonas Mueller and Aditya Thyagarajan. 2016. Siamese Recurrent Architectures for Learning Sentence Similarity.. In *AAAI*, Vol. 16. 2786–2792.

[15] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 311–318.

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

[17] Anukarsh G Prasad, S Sanjana, Skanda M Bhat, and BS Harish. 2017. Sentiment analysis for sarcasm detection on streaming short text data. In *2017 2nd International Conference on Knowledge Engineering and Applications (ICKEA)*. IEEE, 1–5.

[18] Jiliang Tang, Yi Chang, and Huan Liu. 2014. Mining social media with social theories: a survey. *ACM Sigkdd Explorations Newsletter* 15, 2 (2014), 20–29.

[19] Sanjaya Wijeratne, Lakshika Balasuriya, Amit Sheth, and Derek Doran. 2017. A semantics-based measure of emoji similarity. In *Proceedings of the International Conference on Web Intelligence*. ACM, 646–653.

[20] Matthew D Zeiler. 2012. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* (2012).

[21] Zongkui Zhu, Zhengqiu He, Ziyi Tang, Baohui Wang, and Wenliang Chen. 2018. A Semantic Similarity Computing Model based on Siamese Network for Duplicate Questions Identification.. In *CCKS Tasks*. 44–51.