SARCASM RELATION TO TIME: SARCASM DETECTION WITH TEMPORAL FEATURES AND DEEP LEARNING

[1]MD SAIFULLAH RAZALI, [2]ALFIAN ABDUL HALIN, [3]YANG-WAI CHOW, [4]NORIS MOHDNOROWIAND, [5]SHYAMALA DORAISAMY

[1,2,4,5]Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Jalan Universiti 1, Serdang, 43400,Selangor, Malaysia.
[3]Faculty of Engineering and Information Science, University ofWollongong, Northfields Avenue, Wollongong, NSW 2522, New South Wales, Australia.
E-mail: [1]gs47497@student.upm.edu.my, [2]alfian@upm.edu.my, [3]caseyc@uow.edu.au, [4]noris@upm.edu.my, [5]shyamala@upm.edu.my

Abstract - This paper discusses a framework used to detect sarcasm in relation to time. It uses a set of deep learning extracted features (deep features) combined with a set of handcrafted features. The results of the experiments are positive in terms of Accuracy, Precision, Recall and F1measure. The combination of features is classified using a few machine learning techniques for comparison purposes. Logistic Regression is found to be the best classification algorithm for this task with an accuracy of 89%. Furthermore, result comparison to recent works and the performance of each feature set are also shown as additional information.

Keywords - Sarcasm Detection, Natural Language Processing, Deep Learning, Sentiment Analysis.

I. INTRODUCTION

The vast amount of data created on social media today is good for data analysis since they are very personal [1]. Organizations have been using these data to help them understand their audience better [2]. This field is called sentiment analysis [3].

On a different note, sarcasm is defined as a positive sentence with negative intention [4]. It is regarded as one of the most challenging issues in the Natural Language Processing (NLP) field [5]. Handling sarcasm correctly is crucial since it can change the polarity of a sentence [5, 6] and jeopardize a sentiment analysis. Traditional studies [4, 7] used rule-based techniques to solve this and more recent studies [8, 9] used learning to automatically detect the most useful features.

In this paper, the deep features are the product of a deep learning architecture combined with the temporal features which are the product of manual handcrafting. Tweets are used as the main source of input. Usually, different tones and gestures are used in normal day-to-day conversations. These features are not available in writing. Hence, writers that use sarcasm tend to be more creative. They came up with semantic clues to show their sarcasm in a given sentence [4, 7]. This kind of clue is what this work is trying to manipulate.

II. RELATED WORK

Recent works in this domain use deep learning architectures [8, 9]. following its good reputation in solving NLP problems. For example, Poria et al. [8] used four datasets to extract four deep feature sets using Convolutional Neural Network (CNN). Then these feature sets are combined and classified by an SVM classifier. Another work from Ili´c et al. [13] used a deep learning model based on character-level word representations derived from the Embeddings from Language Models (ELMo). ELMo is a representation technique which uses vectors derived from a bidirectional Long Short Term Memory (LSTM) [14]. Both works used the same dataset [15]. This is also the dataset that is used in our work.

On the other hand, most older studies use rules to produce features. For example, Barbieri et al. [16] use frequency and rarity of words as their main features. This technique is also used by Bouazizi and Ohtsuki [17] with additional rules by extracting sarcastic word patterns. The rules include counting the number of positive/negative words in the tweet and counting the number of highly emotional positive/highly emotional negative words. Another study by Shmueli et al [18] used the seed phrase ”being sarcastic” as in ”I was being sarcastic”. The seed is then used to collect other sarcastic instances. Ultimately, a new dataset is created to help solve the issue of dataset scarcity for sarcasm detection.

One of the most prominent studies is this domain by Riloff et al. [4] also used classifiers that look for positive verbs found with negative situations. This research has inspired other researchers that use rules [8, 12, 19]. Apart from rules and deep learning, other studies depended on user historical tweets to create their features. This idea was first experimented by [12] with thorough analysis on what are the most relevant features in a thread of tweets from the same user. The features is as shown in Figure 1. All the features have their own rules. For example, audience feature use historical communication between the

author of the tweet and the person that tweet is intended to. This idea is inspired by the work by Kreuz&Caucci [20] which claims that sarcasm is likely to happen between people that are familiar with each other. This technique is followed by a few other more recent studies [21–23].

For example, Rajadesingan et al. [23] created a framework called Sarcasm Classification Using a Behavior al modeling Approach (SCUBA) where they classify user's behaviour using a similar approach as done by Bamman& Smith [12](using historical tweets) but added aspects like the difference in the length of the words between the user's current tweet and their past tweets. They have also experimented on whether there is a direct correlation between the availability of historical information with sarcasm detection performance, as shown in Figure 2.

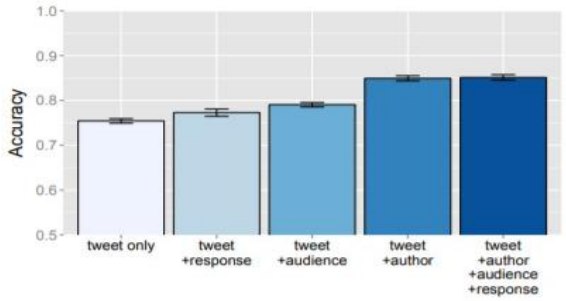


Fig. 1 Most Relevant Features for Sarcasm Detection according to Bamman &Smith[12]

In this work, the deep feature set is combined with a handcrafted set called temporal set. This second set is created after a thorough analysis on an assumption given by another work in the same domain [24]. The author of this work assumes that sarcastic sentences could be regarded as ill-meaning in a period of time but change to be good-meaning in a different period of time. The result

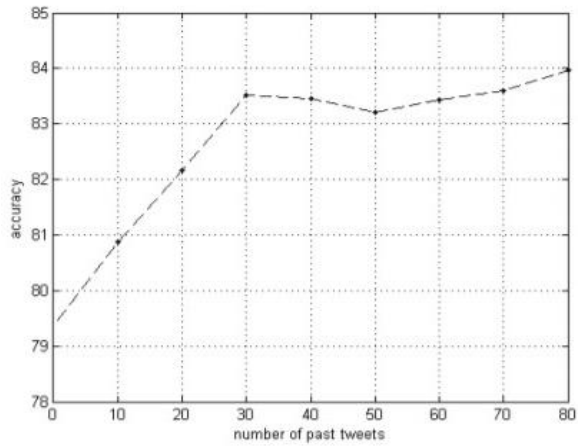


Fig. 2 Direct Correlation between Availability of Historical Information and Sarcasm Detection Performance according to Rajadesinganet al.[23]of the experiments in this work shows that the assumption by Ebrahimi et al. [24] could be deemed right.

III. PROPOSED METHOD

This work with data acquisition and ends with evaluation. When a person is speaking using sarcasm, they tend to use different abnormal tones [25, 26] or exaggerations [27]. In a written statement, these are translated into certain type writings. This work is focusing on the detection of such features. A temporal way of writing is one of those features. This is proven by our experiments in the later sections.

3.1 Data Acquisition

The dataset used as the input for the experiments in this paper is shared publicly [15]. It is an imbalanced dataset of 780,000 English tweets (130,000 sarcastic and 650,000 non-sarcastic). It is imbalance because the author believes that sarcastic utterances rarely happen in the real world in comparison to normal ones. The dataset is split into 80 percent for training and 20 percent for testing.

IV. SARCASM DETECTION

4.2.1 Deep Learning Extraction

Many recent researches on sarcasm detection are moving towards deep learning [8, 9, 22, 28] given its high reputation in NLP. The biggest advantage of using deep learning is its ability to automatically gather optimum features for a given task [8, 9].

In this work, a vanilla Convolutional Neural Network architecture is proposed to extract ten deep features. The overall architecture is shown in Figure 3.

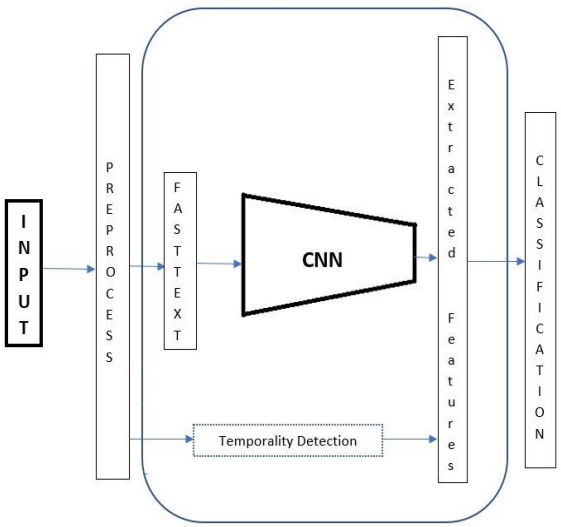


Fig. 3Overall Architecture of Sarcasm Detection Part

As shown in Figure 3, the deep features extractor (CNN) is a large part of the overall architecture of the Sarcasm Detector. It is also assisted by a word embedding technique to convert tweet sentences into feature vectors as input. The details are described below.

4.2.2 FASTTEXT

We have decided to use FastText [29] as our word embedding technique instead of the commonly used Word2Vec [30]. This is because FastText breaks every word into N-grams instead of using individual words as with Word2Vec. Then the words are fed it into a neural network.

FastText feeds the words into a neural network in the form of unigram, bigram and trigram. For example, the word "human" is broken into "hum", "uma" and "man" as a trigram for the word "human". The vectors for "human" will be the total of all the broken N-grams. Artificial Neural Network (ANN) is used as the training method.

The output is a word-embedding vector for all the broken N-grams in the training dataset. Hence, FastText gives a better representation even for the rare and misspelled words. This makes it very effective for social networks analysis.

4.2.3 CONVOLUTIONAL NEURAL NETWORK (CNN)

For the purpose of extracting the deep features, a vanilla CNN architecture as shown in Figure 4 is used. This is the detail of the CNN part from the overall architecture of the sarcasm detection part shown in Figure 3.

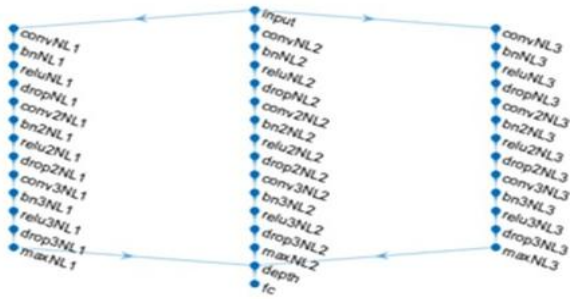


Fig. 4CNN architecture used in the Sarcasm Detection Part

The CNN architecture in Figure 5 is shown in a top-down manner starting from the start (top) to the finish (bottom) node. "NL" stands for N-gram Length. The breakdown is:

1. An input layer of size 1 X 100 X N where N is the number of instances from the dataset. Vectors of embedded-words are used as the initial input.
2. Then the layers between the input and the concatenation is introduced: • One convolutional layer with 200 neurons to receive and filter size 1 X 100 X N where N is the number of instances from the dataset. The stride is [1 1].

- Two convolutional layer with 200 neurons to receive and filter size 1 X 100 X 200. The stride is [1 1].
- Three batch normalization with 200 channels.
- Three ReLU activation layers.
- Three dropout layers with 20 percent dropout.
- A max pooling layer with stride [1 1].

3. A depth concatenation layer to concatenate all the last max pooling layers.
4. A fully connected layer with ten neurons.

4.3 TEMPORALITY DETECTION

Another quality of sarcasm is temporality [24]. For example, the sentence "You think just like Donald Trump now" in 2016 after he won the US election would have a different connotation then in 2020 when he lost the US election. The existence of a temporal word in each tweets is taken as one feature. The checking of whether there is more than one temporal word in each tweets is another feature. Three more features are added by counting the cases where the word that comes in the vicinity of three words before or after the temporal word is a noun, depending on the distance. All of these features are in the form of Boolean. In total, 5 features are extracted using manual handcrafting. For the purpose of extracting them, the lexicons of temporal words and nouns are used.

V. RESULTS AND DISCUSSION

In this section, the results produced by all the experiments is given. This is followed by the explanation of how each component in the experiments are used to ful fill the objective of the the study.

5.1 Classification Results

This subsection consist of performance results for every classification algorithm in the experiment using all the feature sets combined. It is shown in terms of F1-measure, Precision, Recall and Accuracy.

Classification Algorithm	F1	Precision	Recall	Accuracy
SVM	0.87	0.87	0.87	87%
KNN	0.86	0.86	0.86	86%
LR	**0.89**	**0.90**	**0.89**	**89%**
DT	0.87	0.87	0.87	87%
DISCR	0.87	0.87	0.86	86%

Table 1 Performance comparison for Classification Algorithms using all the Feature Sets combined

SVM and Decision Tree are both good classifiers for this task, with high accuracies and F1-scores. However, the performance of Logistic Regression are the highest. For the rest of the analysis in this paper, the results used are those from the classifier Logistic Regression.

5.2 Performance comparison with Existing Works

For comparison, two recent works that used the same dataset as in our experiment are chosen. Both of the chosen works are in the domain of sarcasm detection. The results are compiled and shown in Table 2.

Method	F1	Precision	Recall	Accuracy
Ili´cet al.[13]	0.87	0.87	0.87	88%
Shmueliet al.[18]	0.86	0.87	**0.91**	87%
Proposed Method	**0.89**	**0.90**	0.89	**89%**

Table 2 Performance comparison with Existing Works

The proposed method showed significantly better performance in comparison to the others almost across all metrics used. This supports our claim that manually extracted contextual feature sets are useful for this task. The performance for each of the feature sets is also experimented.

Compared to deep feature, temporal feature shows a lower performance. According to observation, the first reason for this to happen is this type of feature has a low presence in the data set. Secondly, due to the informal language used on Twitter, temporal words are becoming hard to detect. However, an accuracy of more than 60 percent shows its importance. It can be seen clearly from the overall results that the temporal features add value to the whole framework.

VI. CONCLUSION AND FUTURE WORK

The method significantly improves the metrics from the existing study using the same dataset. This work also demonstrates the generality of deep learning architecture. For future work, a few datasets will be used to further generalize the comparisons. The process of extracting and gathering meaningful features could be expanded further.

ACKNOWLEDGMENTS

This research is funded by Universiti Putra Malaysia under the grant: Geran Putra InisiatifSiswazah (Research Title: Sarcasm Detection Using Machine Learning to Enhance Sentiment Analysis).

REFERENCES

[1] Ghosh, A., Veale, T.: Magnets for sarcasm: Making sarcasm detection timely, contextual and very personal. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 482–491 (2017)
[2] Aquino, J.: Transforming social media data into predictive analytics. CRM Magazine 16(11), 38–42 (2012)
[3] Liu, B.: Sentiment analysis and opinion mining. Synthesis lectures on human language technologies 5(1), 1–167 (2012)
[4] Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., Huang, R.: Sarcasm as contrast between a positive sentiment and negative situation. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 704–714 (2013)
[5] Liu, B., Zhang, L.: A survey of opinion mining and sentiment analysis. In: Mining Text Data, pp. 415–463. Springer, ??? (2012)
[6] Joshi, A., Agrawal, S., Bhattacharyya, P., Carman, M.J.: Expect the unexpected: Harnessing sentence completion for sarcasm detection. In: International Conference of the Pacific Association for Computational Linguistics, pp. 275–287 (2017). Springer
[7] Davidov, D., Tsur, O., Rappoport, A.: Semi-supervised recognition of sarcasm in twitter and amazon. In: Proceedings of the Fourteenth Conference on Computational Natural Language Learning, pp. 107–116 (2010)
[8] Poria, S., Cambria, E., Hazarika, D., Vij, P.: A deeper look into sarcastic tweets using deep convolutional neural networks. arXiv preprint arXiv:1610.08815 (2016)
[9] Majumder, N., Poria, S., Peng, H., Chhaya, N., Cambria, E., Gelbukh, A.: Sentiment and sarcasm classification with multitask learning. IEEE Intelligent Systems 34(3), 38–43 (2019)
[10] Abulaish, M., Kamal, A., Zaki, M.J.: A survey of figurative language and its computational detection in online social networks. ACM Transactions on the Web (TWEB) 14(1), 1–52 (2020)
[11] Bouazizi, M., Ohtsuki, T.: Sarcasm detection in twitter:" all your products are incredibly amazing!!!"-are they really? In: 2015 IEEE Global Communications Conference (GLOBECOM), pp. 1–6 (2015). IEEE
[12] Bamman, D., Smith, N.A.: Contextualized sarcasm detection on twitter. In: Ninth International AAAI Conference on Web and Social Media (2015). Citeseer
[13] Ili´c, S., Marrese-Taylor, E., Balazs, J.A., Matsuo, Y.: Deep contextualized word representations for detecting sarcasm and irony. arXiv preprint arXiv:1809.09795 (2018)
[14] Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)
[15] Pt´a˘cek, T., Habernal, I., Hong, J.: Sarcasm detection on czech and english twitter. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 213–223 (2014)
[16] Barbieri, F., Saggion, H., Ronzano, F.: Modelling sarcasm in twitter, a novel approach. In: Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp. 50–58 (2014)
[17] Bouazizi, M., Ohtsuki, T.O.: A pattern-based approach for sarcasm detection on twitter. IEEE Access 4, 5477–5488 (2016)
[18] Shmueli, B., Ku, L.-W., Ray, S.: Reactive supervision: A new method for collecting sarcasm data. arXiv preprint arXiv:2009.13080 (2020)
[19] Joshi, A., Bhattacharyya, P., Carman, M.J.: Automatic sarcasm detection: A survey. ACM Computing Surveys (CSUR) 50(5), 1–22 (2017)
[20] Kreuz, R., Caucci, G.: Lexical influences on the perception of sarcasm. In: Proceedings of the Workshop on Computational Approaches to Figurative Language, pp. 1–4 (2007)
[21] Amir, S., Wallace, B.C., Lyu, H., Silva, P.C.M.J.: Modelling context with user embeddings for sarcasm detection in social media. arXiv preprint arXiv:1607.00976 (2016)
[22] Ghosh, A., Veale, T.: Fracking sarcasm using neural network. In: Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp. 161–169 (2016)
[23] Rajadesingan, A., Zafarani, R., Liu, H.: Sarcasm detection on twitter: A behavioralmodeling approach. In: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, pp. 97–106 (2015)
[24] Ebrahimi, M., Yazdavar, A.H., Sheth, A.: Challenges of sentiment analysis for dynamic events. IEEE Intelligent Systems 32(5), 70–75 (2017)
[25] Attardo, S., Eisterhold, J., Hay, J., Poggi, I.: Multimodal markers of irony and sarcasm. Humor16(2), 243–260 (2003)
[26] Rockwell, P.: Vocal features of conversational sarcasm: A comparison of methods. Journal of psycholinguistic research 36(5), 361–369 (2007)

[27] Kunneman, F., Liebrecht, C., Van Mulken, M., Van den Bosch, A.: Signaling sarcasm: From hyperbole to hashtag. Information Processing & Management 51(4), 500–509 (2015)
[28] Zhang, M., Zhang, Y., Fu, G.: Tweet sarcasm detection using deep neural network. In: Proceedings of COLING 2016, The 26th International Conference on Computational Linguistics: Technical Papers, pp. 2449–2460 (2016)
[29] Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics 5, 135–146 (2017)
[30] Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
[31] Vapnik, V., Izmailov, R.: Knowledge transfer in svm and neural networks. Annals of Mathematics and Artificial Intelligence 81(1), 3–19 (2017)
[32] Peterson, L.E.: K-nearest neighbor. Scholarpedia4(2), 1883 (2009)
[33] Christmann, A.: Least median of weighted squares in logistic regression with large strata. Biometrika81(2), 413–417 (1994)
[34] Kamin´ski, B., Jakubczyk, M., Szufel, P.: A framework for sensitivity analysis of decision trees. Central European journal of operations research 26(1), 135–159 (2018)
[35] Garson, G.D.: Discriminant function analysis. statnotes: topics in multivariate analysis. Retrieved March 29, 2010 (2008)
[36] Huang, J., Ling, C.X.: Using auc and accuracy in evaluating learning algorithms. IEEE Transactions on knowledge and Data Engineering 17(3), 299–310 (2005)

★★★