Multi-task Learning Using a Combination of Contextualised and Static Word Embeddings for Arabic Sarcasm Detection and Sentiment Analysis

Abdullah I. Alharbi
School of Computer Science
University of Birmingham, UK
King Abdulaziz University, KSA
aia784@cs.bham.ac.uk

Mark Lee
School of Computer Science
University of Birmingham, UK
m.g.lee@cs.bham.ac.uk

Abstract

Sarcasm detection and sentiment analysis are important tasks in Natural Language Understanding. Sarcasm is a type of expression where the sentiment polarity is flipped by an interfering factor. In this study, we exploited this relationship to enhance both tasks by proposing a multi-task learning approach using a combination of static and contextualised embeddings. Our proposed system achieved the best result in the sarcasm detection subtask (Abu Farha et al., 2021).

1 Introduction

Social media platforms, such as Twitter, provide the public with an opportunity to share their thoughts and feelings with others through short posts of text. This has created a rich area of research to examine for linguistic expressions used on social media and explore expressions of emotion (sarcasm, humour, offense, etc.). 'Sarcasm' is defined as a figurative type of language where the expression is meant to communicate the opposite of its literal meaning. Within SA, detecting sarcasm is vital as sarcasm typically implies a negative feeling although the positive language is used. This conflict between what is said and what is meant creates a complicated challenge for SA tasks (Bouazizi and Otsuki Ohtsuki, 2016).

Compared to English, only a few studies have been made on Arabic sarcasm detection. Among the studies completed in this area are research by Karoui et al. (2017) and a shared task study by Ghanem et al. (2019). Recently, efforts have been made by Abu Farha and Magdy (2020) and Abbes et al. (2020) to create standard datasets to support sarcasm detection. In the WANLP-2021 Workshop, a shared task ('Sarcasm and Sentiment Detection in Arabic') (Abu Farha et al., 2021) was organised to contribute to the development of this area. Two sub-tasks are included in the shared task. The first will be to identify whether a tweet is sarcastic or not, and the second will be to determine if the tweet is expressing a neutral, positive or negative feeling.

Some substantial difficulties are presented by the shared task. Specifically, the targeted classes are distributed in an unbalanced way and the tweets used are in various dialects and even sub-dialects. Arabic dialects can be significantly different and they lack the established language rules found in Modern Standard Arabic (MSA). To deal with these difficulties, we propose a combined approach that uses three models. By using this combination, we believe we will be able to benefit from the strengths of each model and overcome their weaknesses.

The remainder of this paper is organised as follows. Section 2 comprises an overview of the methodology for our proposed system. Section 3 describes the dataset for the shared task. Section 4 provides experimental results, including the experimental set-up though which the proposed system was tested and a discussion of the outcomes. Finally, Section 5 concludes the paper with suggestions for future research.

2 Method

As described above, our proposed methodology consists of three models. These are the Static Word Embeddings (SWE) , Contextualized Embeddings (MARBERT) and Multi-Task Learning (MTL) models. The following three subsections explain each of these and how our proposed system incorporates them.

2.1 Pre-processing

We adopted the pre-processing techniques that have been undertaken and described by several researchers (Abu Farha and Magdy, 2019; Duwairi and El-Orfali, 2014). First, any unknown symbols

Proceedings of the Sixth Arabic Natural Language Processing Workshop, pages 318–322
Kyiv, Ukraine (Virtual), April 19, 2021.

or non-Arabic characters were removed (e.g. punctuation marks, diacritics, etc.). However, emojis were kept as we believe that they are important for sarcasm and sentiment classification tasks. Several letters that appeared in different forms in the original tweets were normalised. Thereafter, they were rendered into a single form. For instance, the 'hamza' on characters {أ,إ,آ} was replaced with the {ا}, and the 't marbouta' {ة} was replaced with {ه}.

2.2 Static Word Embeddings (SWE)

One primary approach recently applied to the tasks of NLP is word embedding (Zhang et al., 2014; Devlin et al., 2014; Bordes et al., 2014; Lin et al., 2015). Word embedding involves the representation of text as dense vectors and captures semantic similarities between words by positioning them closely together within vectors. Each word is encoded in a real-valued vector, which features hundreds of dimensions. In our work, we used two different level of word embeddings, word-level and character-level embeddings.

Word-level Embeddings: We used a pre-trained word embedding model for Arabic: Ara2Vec (Soliman et al., 2017). The Ara2Vec word embeddings is a well-known open-source tool made up of six Arabic models. Data was obtained from Wikipedia, Twitter and Common Crawl (webpage crawl data) to train Ara2Vec. Although it a significant number of words were used to train Ara2Vec, it is not possible for such a word-level model to recognise every word used in real life. This results in the out-of-vocabulary (OOV) problem, where such resources cannot identify rare words. This is a major shortcoming of word-level embedding models.

Character-level Embeddings: The large number of Arabic dialects spoken in the Arab world can produce worse OOV problems compared to other languages. As a result, resources need to have a better understanding of Arabic dialects to be useful for sarcasm detection and SA tasks. We have found that word-level embeddings offer more value when it comes to semantic similarities, but character-level embeddings are better able to encode word morphology variants and position them more closely in the vector spaces. Thus, we used a pre-trained three-gram character representation model (CE) proposed by Alharbi and Lee (2020), which obtained competitive results in Arabic affect tasks. CE has been generated by training FastText (Bojanowski et al., 2017) on a large dataset of 10 million tweets. This corpus includes various affect words (sentiments and emotions, with different level of intensity) in different Arabic dialects. Table 1 shows a summary of important information about each of these models.

As a baseline, we employed the Convolutional Neural Network (CNN) proposed by Kim (2014). This deep learning architecture separately used one of the aforementioned static word embeddings (AraVec or CE) for initialising the embedding layer weights. These weights were then updated during the training process so they would be more appropriate for the task. This was followed by the application of different filters and kernels in order to generate a collection of features that were max pooled. Finally, these features were passed to the fully connected softmax layer for the classification task. In addition to the CNN network, we also used two different deep learning architectures: Long Short Term Memory (LSTM) network and combining CNN with LSTM network, following the proposed method by Wang et al. (2016). These algorithms were integrated with other models, which are detailed in the following subsections.

Model	level	Domain	Size
Ara2Vec	word	General	77M tweets
CE	character	Emotion	10M tweets

Table 1: The static word embeddings utilised for our systems.

2.3 Contextualized Embeddings Model (MARBERT):

Given the importance of context to determining whether an expression is intended to be sarcastic or not, we decided to use contextualised language models to help us with the shared task. One of these models is the Bidirectional Encoder Representation from Transformer (BERT) model, created by the Google research team, which has had great success with a range of NLP tasks (Devlin et al., 2019). As the dataset of this shared task contains different Arabic dialects, we use a model designed especially for tasks in Arabic dialects, the MARBERT model (Abdul-Mageed et al., 2020). MARBERT has been pre-trained on a significant dataset that includes 6B tweets, which achieves state-of-the-art results in

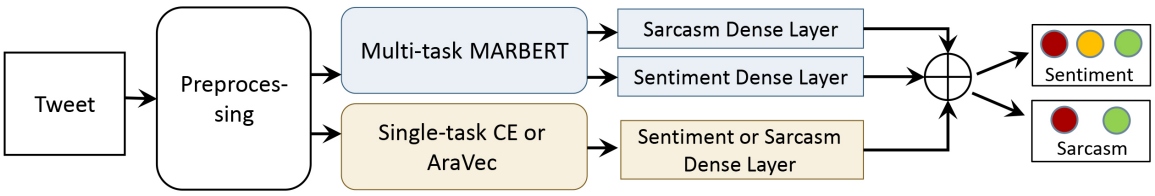

Figure 1: The proposed system architecture.

different Arabic-language NLP tasks. This model uses 12 attention headers, 12 encoder blocks and 768 hidden dimensions. It can handle sequences up to a maximum of 512 tokens, which, when entered into the model, generate a representation of the sequence. The first sequence token is the [CLS] , which contains targeted classification embedding.

We used MARBERT for the single task as a whole following the same implementation as that provided by the authors to target each subtask alone. In addition, we use the sentence embeddings of MARBERT in multi-task learning, as explained below.

2.4 Multi-task Learning (MTL) Model:

Multitask learning is an approach to inductive transfer that enhances generalisation by using the domain knowledge found in the training signals of similar tasks as an inductive bias. The intuition behind MTL is that a useful feature for one task will be useful and thus predictive for other, similar tasks (Caruana, 1997). At present, most research focuses on either detecting sarcasm or classifying sentiments. There has been little research into the idea that these two elements influence each other (Majumder et al., 2019; Zhang and Abdul-Mageed, 2019). We exploit this relationship to enhance both tasks (sarcasm detection and sentiment classification).

We propose a system that integrates an MTL approach with a combination of the aforementioned models (SWE and MARBERT). Figure 1 illustrates the architecture of the proposed method. To combine static and contextualised word embeddings, we followed Alghanmi et al. (2020) by simply concatenating representations obtained from MARBERT and SWE. However, our system enhances this combination of representations by additionally exploiting MTL in both tasks. Furthermore, in order to overcome issues with OOV, we use n-gram character word embeddings instead of just word-level embeddings.

To this end, we extracted the contextualised vectors (average of 12 hidden layers) after fine-tuning the model on the train dataset for each task. The dimension of these vectors after concatenation was 1536 (768 dimensions for each task). For SWE, we obtained the tweet vectors by averaging the real-value word vectors derived from the pre-trained models as a single task. our assumption in only using a single task for SWE instead of MTL is that to provide the target task with more weight. These static vectors were then used as input features into a deep learning architecture, as explained above. At this step, these two types of tweet representations (multi-task MARBERT and single-task SWE) were concatenated. To avoid overfitting, a dropout layer with a 0.5 drop rate was applied. Finally, a dense layer of number-of-classes output was set in place through the exploitation of softmax as an activation function to produce the final output.

3 Data Description

The shared task's organisers released the train and test datasets (ArSarcasm-v2) for both tasks, namely, subtask 1 - sarcasm detection and subtask 2 - sentiment analysis (Abu Farha et al., 2021). The train dataset contained 12,548 tweets while the test dataset consisted of 3,000 tweets. For subtask 1, the tweets were manually annotated using the term 'True' for sarcastic tweets and 'False' for tweets that were not sarcastic. For subtask 2, the tweets were designated as 'NEG' for negative class, 'NEU' for Neutral and 'POS' for positive. An overview of the dataset is provided in Tables 2 and 3.

Dataset	True	False	Total
Training	2168	10380	12,548
Test	821	2179	3000

Table 2: Distribution of classes in Subtask 1.

Dataset	NEG	NEU	POS	Total
Training	4621	5747	2180	12,548
Test	1677	748	575	3000

Table 3: Distribution of classes in Subtask 2.

4 Experiment Results

4.1 Results and Evaluation

Since no labelled development or test dataset was available, we decided to use five-fold cross-validation on the training dataset for evaluation purposes. However, our best model was trained on all training datasets for the final submission for both tasks. The official evaluation metric for subtask 1 was the F-score of the sarcastic class (F1-sarcastic) and the macro average of the F-score of the positive and negative classes (F-PN) for subtask 2.

Table 4 presents the models' performance using the average of cross-validation for both subtasks. For the sarcasm detection task, the best result (0.632) was obtained by the proposed system, which integrates MTL with MARBERT and then combines with SWE in the CNN-LSTM network architecture. From this result, it is evident that the MTL model is important for the sarcasm detection task. In addition, the character-level embeddings (CE) achieve better results compared with the word-level embeddings (AraVec) for the baseline model (CNN). Therefore, we decided to use CE as the static word embeddings for the MTL model. We submitted the predication classes based on the MTL-CNN-LSTM system, which achieved state-of-the-art compared with the other systems proposed by other participants.

For the SA subtask, Table 4 shows the results for all models, where the proposed system obtained the best result on the cross-validation metric. However, using MARBERT as a single-task obtained a very close result based on the five-fold cross-validation evaluation. The MTL-CNN-LSTM system has also been submitted for the SA subtask, which ranked 11th out of 27 participants. We believe this result should be studied as a future work by investigating different setup settings and, more importantly, to analyse errors on the test dataset.

4.2 Submission Results

For both aforementioned subtasks, we (BhamNLP team) submitted the predication classes based on the MTL-CNN-LSTM model. For the sarcasm detection subtask (1), the proposed model achieved state-of-the-art compared with the other systems proposed by other participants. For the SA subtask (2), the model achieved a macro F1-PN of 0.701 on the test set. Table 5 presents the official results achieved by our proposed model (MTL-CNN-LSTM) on the test set for Subtasks 1 and 2.

Model	Sarcasm (F1-sarcastic)	Sentiment (F-PN)
CNN-AraVec	0.486	0.534
CNN-CE	0.498	0.604
MARBERT	0.609	0.702
MTL-CNN	0.603	0.666
MTL-LSTM	0.619	0.698
MTL-CNN-LSTM	**0.632**	**0.713**

Table 4: Performance of the different models using the mean of five-fold Cross-Validation for Subtasks 1 and 2.

Task	Main Metric	Result	Rank
Sarcasm	F1-sarcastic	0.623	1st
Sentiment	F-PN	0.701	11th

Table 5: Main metric results obtained by the proposed model on the test set for both subtasks 1 and 2.

5 Conclusions

In this work, we described an MTL approach using a combination of static and contextualised word embeddings. The proposed system achieved the best result in sarcasm detection subtask and ranked 11th in SA task. In future studies, we will investigate the impact of sarcasm on SA task to enhance the performance of our proposed system for SA task. Additionally, advanced pre-possessing techniques should be applied as we observed some multi-words need to be segmented and treated. Also, instead of removing all punctuation symbols, some of these marks (e.g. question marks and exclamation marks) may be useful especially for the sarcasm detection task.

References

Ines Abbes, Wajdi Zaghouani, Omaima El-Hardlo, and Faten Ashour. 2020. DAICT: A dialectal Arabic irony corpus extracted from Twitter. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6265–6271, Marseille, France. European Language Resources Association.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2020. ARBERT & MARBERT: Deep bidirectional transformers for Arabic.

Ibrahim Abu Farha and Walid Magdy. 2019. Mazajak: An online Arabic sentiment analyser. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 192–198, Florence, Italy. Association for Computational Linguistics.

Ibrahim Abu Farha and Walid Magdy. 2020. From Arabic sentiment analysis to sarcasm detection: The ArSarcasm dataset. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 32–39, Marseille, France. European Language Resource Association.

Ibrahim Abu Farha, Wajdi Zaghouani, and Walid Magdy. 2021. Overview of the WANLP 2021 shared task on sarcasm and sentiment detection in Arabic. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*.

Israa Alghanmi, Luis Espinosa Anke, and Steven Schockaert. 2020. Combining BERT with static word embeddings for categorizing social media. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 28–33, Online. Association for Computational Linguistics.

Abdullah I. Alharbi and Mark Lee. 2020. Combining character and word embeddings for affect in Arabic informal social media microblogs. In *Natural Language Processing and Information Systems*, pages 213–224, Cham. Springer International Publishing.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question Answering with Subgraph Embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 615–620.

M. Bouazizi and T. Otsuki Ohtsuki. 2016. A pattern-based approach for sarcasm detection on twitter. *IEEE Access*, 4:5477–5488.

Rich Caruana. 1997. Multitask Learning. *Machine Learning*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis. Association for Computational Linguistics.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1370–1380.

Rehab Duwairi and Mahmoud El-Orfali. 2014. A study of the effects of preprocessing strategies on sentiment analysis for Arabic text. *Journal of Information Science*, 40(4):501–513.

Bilal Ghanem, Jihen Karoui, F. Benamara, Véronique Moriceau, and P. Rosso. 2019. Idat at fire2019: Overview of the track on irony detection in Arabic tweets. *Proceedings of the 11th Forum for Information Retrieval Evaluation*.

Jihen Karoui, Farah Banamara Zitoune, and Véronique Moriceau. 2017. SOUKHRIA: Towards an irony detection system for Arabic in social media. *Procedia Computer Science*, 117:161–168. Arabic Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Chu-Cheng Lin, Waleed Ammar, Chris Dyer, and Lori Levin. 2015. Unsupervised POS Induction with Word Embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1311–1316.

N. Majumder, S. Poria, H. Peng, N. Chhaya, E. Cambria, and A. Gelbukh. 2019. Sentiment and sarcasm classification with multitask learning. *IEEE Intelligent Systems*, 34(3):38–43.

Abu Bakr Soliman, Kareem Eissa, and Samhaa R El-Beltagy. 2017. AraVec: A set of Arabic word embedding models for use in Arabic NLP. *Procedia Computer Science*, 117:256–265.

Xingyou Wang, Weijie Jiang, and Zhiyong Luo. 2016. Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2428–2437, Osaka, Japan. The COLING 2016 Organizing Committee.

Chiyu Zhang and Muhammad Abdul-Mageed. 2019. Multi-task bidirectional transformer representations for irony detection.

Jiajun Zhang, Shujie Liu, Mu Li, Ming Zhou, and Chengqing Zong. 2014. Bilingually-constrained phrase embeddings for machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 111–121.