Combining Context-Free and Contextualized Representations for Arabic Sarcasm Detection and Sentiment Identification

Amey Hengle[1], **Atharva Kshirsagar**[1], **Shaily Desai**[1], and **Manisha Marathe**[2]

[1,2]Department of Computer Engineering, PVG's College of Engineering and Technology, affiliated to Savitribai Phule Pune University, India.
[1]{ameyhengle22, atharvakshirsagar145, shailysd02} @gmail.com
[2]mvm_comp@pvgcoet.ac.in

Abstract

Since their inception, transformer-based language models have led to impressive performance gains across multiple natural language processing tasks. For Arabic, the current state-of-the-art results on most datasets are achieved by the AraBERT language model. Notwithstanding these recent advancements, sarcasm and sentiment detection persist to be challenging tasks in Arabic, given the language's rich morphology, linguistic disparity and dialectal variations. This paper proffers team SPPU-AASM's submission for the WANLP ArSarcasm shared-task 2021, which centers around the sarcasm and sentiment polarity detection of Arabic tweets. The study proposes a hybrid model, combining sentence representations from AraBERT with static word vectors trained on Arabic social media corpora. The proposed system achieves a F1-sarcastic score of 0.62 and a F-PN score of 0.715 for the sarcasm and sentiment detection tasks, respectively. Simulation results show that the proposed system outperforms multiple existing approaches for both the tasks, suggesting that the amalgamation of context-free and context-dependent text representations can help capture complementary facets of word meaning in Arabic. The system ranked second and tenth in the respective sub-tasks of sarcasm detection and sentiment identification.

1 Introduction

With the advent of social media platforms as a valuable source of opinion-rich information, work on subjectivity language analysis has continued to receive increasing interest from the Arabic NLP community. Sentiment Analysis (SA) has been the dominant theme in this area, with notable works ranging from the creation of lexical resources and sentiment datasets (El-Beltagy, 2016; Badaro et al., 2014; AbdelRahim Elmadany and Magdy, 2018; Kiritchenko et al., 2016) to the contrivance of neural network-based classification models (Alayba et al., 2018; Heikal et al., 2018; Kaibi et al., 2020). In comparison, the literature in Arabic sarcasm detection is still in its nascent stage, limited to a few notable works (Karoui et al., 2017; Ghanem et al., 2019; Abbes et al., 2020).

Recently, transformer-based language models have proved highly efficient at language understanding, achieving promising results across multiple NLP tasks and benchmark datasets. The language modeling capability of these models is aiding in capturing the literal meaning of context-heavy texts. For Arabic NLP in particular, the best results for sentiment analysis are currently achieved by AraBERT, a language model proposed by Antoun et al. (2020).

Despite this recent progress, sarcasm detection remains a challenging task, primarily due to the use of implicit, indirect phrasing and the figurative nature of language (Abu Farha et al., 2021). The task becomes even more challenging when working with Twitter data, as the social media posts tend to be short and often contain sources of noise, code-switching, and the use of nonstandard dialectal variations (Baly et al., 2017). Furthermore, BERT-based models are found to struggle with rare words (Schick and Schütze, 2019), which can be encountered more in social media texts due to their informal nature and the prevalent use of slang words. For language models like AraBERT, this can pose a challenge, primarily since it has been trained on structured corpora from Wikipedia.

Building on the capabilities of language models, some recent studies have shown that incorporating entity vectors can benefit the BERT-based language models, especially for domain-specific tasks or datasets (Poerner et al., 2020; Lin et al., 2019; Peinelt et al., 2020). An interesting approach followed by Alghanmi et al. (2020) suggests that the

Proceedings of the Sixth Arabic Natural Language Processing Workshop, pages 357–363
Kyiv, Ukraine (Virtual), April 19, 2021.

performance of language models can be boosted by incorporating static word embeddings trained on specific social media corpora.

In this study, we posit a solution for subtask-1 and subtask-2 of the WANLP ArSarcasm shared task 2021 (Abu Farha et al., 2021). While subtask-1 focuses on identifying sarcasm, subtask-2 deals with classifying the sentiment polarity in Arabic tweets. Inspired by the works of Peinelt et al. (2020) and Alghanmi et al. (2020), we propose a hybrid model, combining the sentence representations learned from AraBERT with pre-trained Arabic word vectors proposed by Abu Farha and Magdy (2019). Results attest that the proposed methodology can provide a competent way of subsuming the advantages of both the contextualized and context-free word representations, outperforming all the baseline models for both tasks.

The rest of the paper is organized as follows: In Section 2, we provide a concise literature review of previous works in Arabic sentiment and sarcasm detection. Section 3 provides a descriptive analysis of the dataset at hand. In Section 4, we present the proposed system, and in Section 5, we describe the experimental setup details. Section 6 interprets the results. Finally, Section 7 concludes the study and points to possible directions for future work.

2 Related Work

Early works in Arabic subjectivity analysis focused on using conventional machine learning approaches and lexical methods (Al-Ayyoub et al., 2019). With the emergence of deep learning techniques, research in Arabic NLP has shifted from the traditional statistical standpoint to designing complex neural network models and learning word representations. Al Sallab et al. (2015) experimented with various deep learning models such as a recursive autoencoder (RAE), deep belief networks (DBN), and a deep auto-encoder (DAE). Alayba et al. (2018) built an Arabic SA system based on a combination of CNNs and LSTMs. In (Al-Smadi et al., 2018), the authors proposed an aspect-based sentiment analysis system based on a hybrid architecture of BiLSTM and conditional random field (CRF).

The success of the English word2vec (Mikolov et al., 2013) and fast-Text (Bojanowski et al., 2016) motivated other works to achieve the same feat by creating language-specific word embeddings. For Arabic NLP, some early attempts include word2vec-based AraVec (Soliman et al., 2017), followed by fast-Text (Bojanowski et al., 2016). Recently, Abu Farha and Magdy (2019) proposed the Mazajak embeddings, trained exclusively on a large Arabic twitter corpus for handling the varied Arabic dialects. Multiple studies leveraged this advancement in word representations for both the sarcasm and sentiment detection tasks. For instance, Heikal et al. (2018) developed a CNN and LSTM ensemble model for Arabic SA. The authors employed pre-trained AraVec word embeddings for the text representation. Kaibi et al. (2020) proposed a hybrid model for Arabic SA, concatenating pre-trained AraVec and fast-Text vectors. Abu Farha and Magdy (2019) used the pre-trained Mazajak vectors on a CNN-BiLSTM ensemble model, achieving state-of-the-art results on three benchmark datasets. For sarcasm detection, a similar approach was followed by Abu Farha et al. (2021), where the authors used the Mazajak vectors in combination with a BiLSTM model.

The best results for multiple datasets is currently achieved by fine-tuning the AraBERT model (Antoun et al., 2020), eliminating the need to use the static word vectors in standard settings. Despite this fact, we believe that AraBERT and Mazajak have complementary strengths and can lead to improved results if used in coalescence. In this study, we investigate the effectiveness of combining the word representations obtained from these two models on the sarcasm detection and sentiment identification tasks.

3 Dataset

The WANLP ArSarcasm shared-task 2021 follows the ArSarcasm v2 dataset (Abu Farha et al., 2021). The dataset contains sarcasm, sentiment and dialect labels of 12,549 Arabic tweets. The tweets span across five Arabic dialects including MSA, Gulf, Egyptian, Levantine, and Maghrebi, with MSA and Egyptian dialects having the highest percentage of tweets. For development, we follow a standard 80-20 stratified split on the ArSarcasm v2 dataset. This leaves us with a validation set of 2510 tweets, which are used for the primary evaluation of the proposed system along with the baseline models. The organizers provide a separate dataset for testing, consisting of 3,000 tweets. Table 1 and Table 2 provide a descriptive analysis of the final training, validation and test sets for the tasks of sentiment identification and sarcasm detection respectively.

Set	Positive	Negative	Neutral	Total
Training	1744	3697	4598	10039
Validation	436	925	1149	2510
Testing	575	1677	748	3000
Total	2755	6298	6495	15548

Table 1: Label wise distribution for sentiment analysis.

Set	True	False	Total
Training	1734	8305	10039
Validation	434	2076	2510
Testing	821	2179	3000
Total	2989	12559	15548

Table 2: Label wise distribution for sarcasm detection.

4 Proposed System

Unlike the static word models such as word2vec or fast-Text, language models like AraBERT follow a different tokenization strategy; wherein each word is split into one or more wordpiece tokens (Antoun et al., 2020). Thus, we cannot simply concatenate the pre-trained Mazajak embeddings with the contextualized representations predicted by AraBERT at the word level. We instead combine these word representations at the sentence level, following an approach similar to the one used by Peinelt et al. (2020). The final sentence representation is passed to the dropout layer, followed by a dense layer with Softmax activation for classification. Fig 1 gives an overview of the proposed system's architecture. The following section describes each system component in detail.

4.1 CNN-BiLSTM emsemble

In order to get a sentence representation from the static word embeddings, we employ a CNN-BiLSTM ensemble model. The model learns a 128-dimensional feature vector from the pre-trained Mazajak embeddings. While CNN excels at extracting features from the input data, BiLSTM supports better modeling of sequential dependencies. Thus, using an ensemble helps us subsume the advantages of both these techniques.

4.1.1 Static Embedding Input

The proposed system makes use of the Skip-Gram model variant of the Mazajak word embeddings, pre-trained on 250 million Arabic tweets [1] . During training, the embedding layer maps each word in a tweet to its corresponding embedding, converting the tweet into a real valued dense vector.

[1]http://mazajak.inf.ed.ac.uk:8000/

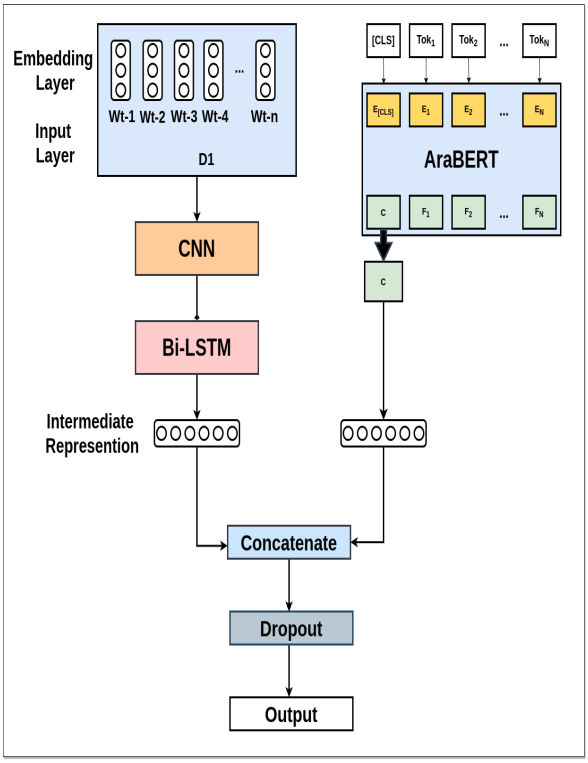

Figure 1: Architecture diagram of the proposed model.

4.1.2 CNN Layer

The word vectors generated by the embedding layer are fed to a CNN layer. The discrete convolutions performed by the CNN layer help to extract the most influential n-grams in the tweet. This is followed by a max-pooling layer.

4.1.3 BiLSTM Layer

The bidirectional short term memory (BiLSTM) layer is just a combination of two LSTMs running in opposite direction (Graves and Schmidhuber, 2005), allowing the network to simultaneously encode the forward and backward information of a sequence at each time step. The intermediate sentence representation generated by the CNN layer is passed to the BiLSTM layer, which encodes it into a 128-dimensional feature vector $D \in \mathbb{R}^{d_1}$.

4.2 AraBERT

The contextualized tweet representations are obtained using the pre-trained AraBERT model (Antoun et al., 2020). In particular, each tweet t is encoded using the C vector from AraBERT's final hidden state layer corresponding to the special

classification token [CLS]:

$$C = AraBERT(t) \in \mathbb{R}^{d_2} \quad (1)$$

where d_2 denotes the internal hidden size of AraBERT (768 for AraBERT v0.2).

Computed using self-attention, the [CLS] token vector is designed to collect information from the rest of the hidden states and be used as a unique representation of the entire sequence (Devlin et al., 2019). We find that this method is more robust than averaging the hidden states, mainly since it avoids every state to be averaged with the same weight, including stopwords and tokens not relevant to the classification task.

4.3 Combined Classification Layer

The final sentence representation F is obtained by simply concatenating the C vector predicted by the AraBERT model with the feature vector D obtained from the CNN-BiLSTM ensemble.

$$F = [D; C] \in \mathbb{R}^{d1+d2} \quad (2)$$

After applying a dropout, the resultant concatenated vector is passed to a dense layer with Softmax activation for classification.

5 Experimental Setup

5.1 Data Preparation

For each tweet in the corpus, we apply standard text cleaning steps including the removal of hashtags, mentions, urls, punctuations and arabic diacritics (Said et al., 2009). For stopword removal, we used a publicly available resource [2]. To replace emojis and emoticons with their corresponding Arabic translations, we created a custom dictionary mapping. Arabic text normalization and lemmatization is done using the AraBERT preprocessor [3]. Each tweet is padded to a maximum length of 100 for both the AraBERT and the CNN-BiLSTM model. Longer tweets are truncated.

5.2 Parameters and Training Environment

For the CNN-BiLSTM ensemble part of the proposed system, we employ a CNN layer with 256 filters and relu activation. For the BiLSTM layer, we use 128 dimensional units (64 for each LSTM), and apply a recurrent dropout rate of 0.2. All layers are implemented using Keras [4]. For using the pre-trained AraBERT model, we follow the Tensorflow implementation of Hugging Face [5]. We make use of the AraBERT v0.2 version, which is officially available under the name bert-base-arabertv02. The same model is used for text tokenization. After predicting the [CLS] token vector from the AraBERT model, we freeze the model weights. We then jointly train the CNN-BiLSTM ensemble with the combined classification layer. The model is trained using the Adam optimizer (Kingma and Ba, 2017), a learning rate of 5e-5, epsilon value of 1e-08, clipnorm equal to 1.0, a batch size of 16, and sparse categorical crossentropy loss, with the usage of early stopping for a callback.

5.3 Baseline Models

As a baseline, we compare the performance of the proposed system against the results of the pre-trained AraBERT model (Antoun et al., 2020), for both the sarcasm and sentiment detection tasks. Furthermore, we show the performance of the previous systems proposed by Abu Farha and Magdy (2020) and Abu Farha and Magdy (2019) for the tasks of sarcasm detection and sentiment identification respectively.

6 Results and Discussion

The official evaluation metric for the sarcasm detection subtask is the F-score of the sarcastic class, while that for the sentiment identification subtask is the F-PN score (macro average of the F-score of the positive and negative classes). For each model, the validation set results are averaged over five runs to ensure a fair comparison.

Table 3 shows the results for the sarcasm detection subtask. The proposed model shows a 10-percent improvement in the F1-sarcasm score over the baseline AraBERT model in identifying sarcastic tweets. This indicates that the proposed system offers a more nuanced ability to capture the figurative meaning of tweets and identify implicit negative sentiments. Table 4 lists the baseline models' performances and the proposed system for sentiment identification subtask. It is observed that both the proposed and AraBERT baseline model perform well on the neutral class. However, a better F-PN score indicates that the proposed model can more efficiently distinguish the positive and negative sentiment polarities from the neutral class. Furthermore, the proposed system seems to

[2] https://github.com/mohataher/arabic-stop-words
[3] https://github.com/aub-mind/arabert
[4] https://keras.io/about/
[5] https://github.com/huggingface/transformers

Model	Accuracy	Precision	Recall	F1-Macro	F1-Sarcastic
(Abu Farha and Magdy, 2020)	-	0.62	0.38	0.46	0.44
AraBERT	0.85	0.75	0.70	0.72	0.52
AraBERT + CNN-BiLSTM	**0.86**	**0.76**	**0.78**	**0.77**	**0.62**
AraBERT + CNN-BiLSTM (Official results on test set)	**0.7410**	**0.7031**	**0.7447**	**0.7096**	**0.6140**

Table 3: Performance comparison of models for subtask-1 : sarcasm detection. All metrics correspond to the results on the sarcastic class.

Model	Accuracy	Precision	Recall	F1-Macro	F-PN
(Abu Farha and Magdy, 2019)	0.67	0.64	0.66	0.64	0.60
AraBERT	0.73	0.71	0.68	0.70	0.67
AraBERT + CNN-BiLSTM	**0.75**	**0.72**	**0.73**	**0.72**	**0.71**
AraBERT + CNN-BiLSTM (Official results on test set)	**0.6840**	**0.6421**	**0.6388**	**0.6232**	**0.7073**

Table 4: Performance comparison of models for subtask-2: sentiment identification.

better handle the data-imbalance for both the tasks and is more robust to overfitting on the minority classes, showing a significant lead in the recall scores.

Overall, the proposed method shows improved results across all the metrics for both the sarcasm and sentiment detection tasks. The performance improvements can be attributed to the fact that unlike the Mazajak word embeddings, which are exclusively trained on a Twitter corpus, the AraBERT model is trained on the Arabic Wikipedia and news corpora, preventing it from witnessing the varied dialects in which social media posts are written. It is reasonable to postulate that while language models like AraBERT capture rich contextual information, the Mazajak word vectors can provide valuable complementary information for rare words, which can be found abundantly in social media texts. Hence, using them in combination can only help a system capture complementary facets of word meaning, thereby enhancing its performance on the downstream sentiment and sarcasm detection tasks.

7 Conclusion

In this study, we proposed a hybrid model to combine the contextualized sentence representations obtained from AraBERT with pre-trained Mazajak word vectors. We show that the proposed model outperforms the standalone AraBERT model for both the sarcasm and sentiment detection tasks. Our findings suggest that incorporating static word vectors might help language models like AraBERT to deal with rare words and the constantly updating language of social media. An alternative strategy would be to pre-train AraBERT on specific social media corpora like Twitter. However this can prove to be extremely expensive and is not feasible in practice. It is also important to note that, while the proposed model leads in performance, it is more complex, and has a greater number of trainable parameters. Hence, it would be essential to test its feasibility on datasets larger than the ArSarcasm v2 dataset.

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.

Hamdy Mubarak AbdelRahim Elmadany and Walid Magdy. 2018. Arsas: An arabic speech-act and sentiment corpus of tweets. In *Proceedings of the Eleventh International Conference on Language*

Resources and Evaluation (LREC 2018), Paris, France. European Language Resources Association (ELRA).

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*.

Mahmoud Al-Ayyoub, Abed Allah Khamaiseh, Yaser Jararweh, and Mohammed N Al-Kabi. 2019. A comprehensive survey of arabic sentiment analysis. *Information processing & management*, 56(2):320–342.

Ahmad Al Sallab, Hazem Hajj, Gilbert Badaro, Ramy Baly, Wassim El Hajj, and Khaled Bashir Shaban. 2015. Deep learning models for sentiment analysis in Arabic. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 9–17, Beijing, China. Association for Computational Linguistics.

Mohammad Al-Smadi, Omar Qawasmeh, Mahmoud Al-Ayyoub, Y. Jararweh, and Brij Gupta. 2018. Deep recurrent neural network vs. support vector machine for aspect-based sentiment analysis of arabic hotels' reviews. *J. Comput. Sci.*, 27:386–393.

Abdulaziz M. Alayba, Vasile Palade, Matthew England, and Rahat Iqbal. 2018. A combined cnn and lstm model for arabic sentiment analysis. *Machine Learning and Knowledge Extraction*, page 179–191.

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.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Gilbert Badaro, Ramy Baly, Hazem Hajj, Nizar Habash, and Wassim El-Hajj. 2014. A large scale Arabic sentiment lexicon for Arabic opinion mining. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 165–173, Doha, Qatar. Association for Computational Linguistics.

Ramy Baly, Gilbert Badaro, Georges El-Khoury, Rawan Moukalled, Rita Aoun, Hazem Hajj, Wassim El-Hajj, Nizar Habash, and Khaled Shaban. 2017. A characterization study of Arabic Twitter data with a benchmarking for state-of-the-art opinion mining models. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pages 110–118, Valencia, Spain. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Samhaa R. El-Beltagy. 2016. NileULex: A phrase and word level sentiment lexicon for Egyptian and Modern Standard Arabic. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2900–2905, Portorož, Slovenia. European Language Resources Association (ELRA).

Bilal Ghanem, Jihen Karoui, Farah Benamara, Véronique Moriceau, and Paolo Rosso. 2019. Idat at fire2019: Overview of the track on irony detection in arabic tweets. In *Proceedings of the 11th Forum for Information Retrieval Evaluation*, pages 10–13.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610.

M. Heikal, Marwan Torki, and N. El-Makky. 2018. Sentiment analysis of arabic tweets using deep learning. In *ACLING*.

Ibrahim Kaibi, Hassan Satori, et al. 2020. Sentiment analysis approach based on combination of word embedding techniques. In *Embedded Systems and Artificial Intelligence*, pages 805–813. Springer.

Jihen Karoui, Farah Banamara Zitoune, and Veronique Moriceau. 2017. Soukhria: Towards an irony detection system for arabic in social media. *Procedia Computer Science*, 117:161–168.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Svetlana Kiritchenko, Saif Mohammad, and Mohammad Salameh. 2016. SemEval-2016 task 7: Determining sentiment intensity of English and Arabic phrases. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 42–51, San Diego, California. Association for Computational Linguistics.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning.

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

Nicole Peinelt, Dong Nguyen, and Maria Liakata. 2020. tBERT: Topic models and BERT joining forces for semantic similarity detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7047–7055, Online. Association for Computational Linguistics.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. E-BERT: Efficient-yet-effective entity embeddings for BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 803–818, Online. Association for Computational Linguistics.

Dina Said, Nayer M Wanas, Nevin M Darwish, and Nadia Hegazy. 2009. A study of text preprocessing tools for arabic text categorization. In *The second international conference on Arabic language*, pages 230–236.

Timo Schick and Hinrich Schütze. 2019. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking.

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. Arabic Computational Linguistics.