Proceedings of the OSACT 2022 Workshop @LREC2022, pages 209–213
Marseille, 20 June 2022

GUCT at Arabic Hate Speech 2022: Towards a Better Isotropy for Hate Speech Detection

Nehal Elkaref, Mervat Abu-Elkheir
German University in Cairo
nehal.elkaref@student.guc.edu.eg, mervat.abuelkheir@guc.edu.eg

Abstract

Hate Speech is an increasingly common occurrence in verbal and textual exchanges on online platforms, where many users, especially those from vulnerable minorities, are in danger of being attacked or harassed via text messages, posts, comments, or articles. Therefore, it is crucial to detect and filter out hate speech in the various forms of text encountered on online and social platforms. In this paper, we present our work on the shared task of detecting hate speech in dialectical Arabic tweets as part of the OSACT shared task on Fine-grained Hate Speech Detection. Normally, tweets have a short length, and hence do not have sufficient context for language models, which in turn makes a classification task challenging. To contribute to sub-task A, we leverage MARBERT's pre-trained contextual word representations and aim to improve their semantic quality using a cluster-based approach. Our work explores MARBERT's embedding space and assess its geometric properties in-order to achieve better representations and subsequently better classification performance. We propose to improve the isotropic word representations of MARBERT via clustering. we compare the word representations generated by our approach to MARBERT's default word representations via feeding each to a bidirectional LSTM to detect offensive and non-offensive tweets. Our results show that enhancing the isotropy of an embedding space can boost performance. Our system scores 81.2% on accuracy and a macro-averaged F1 score of 79.1% on sub-task A's development set and achieves 76.5% for accuracy and an F1 score of 74.2% on the test set.

Keywords: hate speech, isotropy, contextual word embeddings, pre-trained-models, representation degeneration problem

1. Introduction

Social media platforms are continuously being used as a medium for freedom of expression. However, users tend to abuse this liberty and disregard the possibility of their comments harbouring any sort of hate or offensive content. Hatespeesh could be considered as an umbrella-term for various expressions that can be offensive interms of one's gender, race, religion, disability, or ethnicity or simply encourage hatred towards certain groups and individuals. Such expressions can elevate to threats and can put users in the risk of being cyberbullied. Therefore, tackling the issue of classifying online content as containing any sort of hate is crucial to users safety.

2. Related Work

Previous research efforts and methods to detect hatespeech within the scope of Arabic language include a comparative study between word representations, distributed term representation and statistical bag of words (Abuzayed and Elsayed, 2020) where they showcased that the former outperforms the latter in a joint CNN and LSTM architecture. In the same vein (Saeed et al., 2020) compared between embeddings that support the Arabic language, amongst them are Word2Vec (Church, 2017), FastText (Joulin et al., 2016), BERT's multilingual vectors[1], and AraVec (Soliman et al., 2017). Their evaluation yielded Word2Vec and FastText as better suited for the task. They concatenated both embeddings and used deep learning models for the learning process in addition to a stack of classifiers fine-tuned for classification. Similarly, the study in (Saeed et al., 2020) explored the use of different word embeddings including Word2Vec, FastText and GloVe accompanied by different neural network models, those of which included LSTM, CNN and GRU, to find the best performing combination of word embeddings and neural network architecture. The research in (Husain, 2020) showed that substantial preprocessing of Arabic data could remarkably escalate performance, where they converted emojis in texts into their corresponding label, thus creating additional features. Moreover, they attempted to reduce dimensionality by neutralising the dialectical nature of the dataset and turning it into Modern Standard Arabic (MSA), in addition to the other basic cleaning and preprocessing techniques.

All the aforementioned systems aimed to give better representations either using different kind of embeddings or extensive preprocessing, and in the same vein to their work, we tackle this shared task by using and refining MARBERT's (Abdul-Mageed et al., 2020) contextual word representations. To give a general framework of our method, we firstly enhance MARBERT's embedding space by making it more isotropic, then we extract contextual representations from the improved geometric space to finally feed them to a Bidirectional LSTM to detect offensive tweets. We compare the performance of isotropic representations

[1] https://github.com/google-research/bert/blob/master/multilingual.md

against anisotropic or default representations of MARBERT.

3. Background

An isotropic embedding space is a space where all representations are scattered uniformly in all directions. Having this geometric property enhances expressiveness in the embedding space, enables an optimizer algorithm's an improved convergence rate(Rajaee and Sheikhi,), and improves overall performance.(Ioffe and Szegedy, 2015)(Wang et al., 2019) (Rajaee and Pilehvar, 2021a).

Multiple research efforts have shown that architectures that utilise contextual and non-contextual word representations lack isotropy (i.e anisotropic) (Devlin et al., 2018) (Rajaee and Pilehvar, 2021b)(Rajaee and Pilehvar, 2021a). This stems from what is known as the *Representation Degeneration Problem* discovered by (Gao et al., 2019), where they showed that when likelihood maximization is used with the weight tying to train a model on large corpora for language generation tasks, high frequency words are pushed towards hidden states while low frequency words are pushed in the negative direction of most hidden states. This creates a cluster in the embedding space of low frequency words, far away from the origin. This problem impacts the distribution of word vectors as they become dispersed into a cone-like shape making the embedding space *anisotropic*. Such property can negatively impact the training time (Huang et al., 2018) and can overestimate the cosine similarity between representations. (Gao et al., 2019). There have been multiple strategies to tweak embedding spaces to become more isotropic (Devlin et al., 2018) (Liang et al., 2021). We use (Rajaee and Pilehvar, 2021a)'s cluster-based approach in our work to achieve an embedding space with representations homogeneously dispersed.

The rest of the paper is organised as follows, in section 4 we describe the objective of the sub-task and the nature of the data. We also, quantify isotropy and prove numerically and visually that MARBERT has an anisotropic geometry. In section5 we explain the methodology to improving MARBERT's isotropy and the architecture used to learn representations and perform classification. Finally in section6 we reveal our results and findings and discuss additional settings used to attempt score better results.

4. Dataset and Sub-task

The OSACT5 Shared Task[2] consists of three sub-tasks, we contribute to one sub-task (sub-task A) which requires classifying a tweet as offensive or not offensive. The training samples are in dialectical Arabic and consist of 8.5K tweets for training and 1.2K tweets for development collected by (Mubarak et al., 2022). We provide below the number of offensive to non-offensive tweets. The dataset features a number of emojis per data sample that can relay the intention of the tweet as containing hate-speech generally or not.

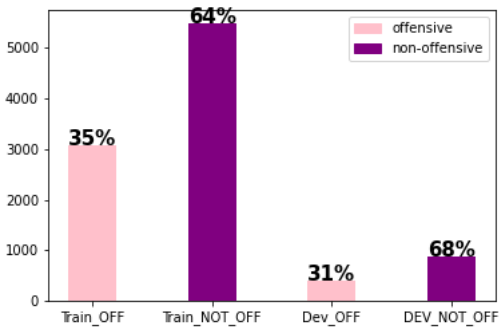

Figure 1: Offensive to non-offensive samples in training and development sets

Data samples are already partially pre-processed; any instances of twitter mentions are replaced with '@USER' and URLs by 'URL'. We remove these tokens along with diacritics, elongations and any instance of non-Arabic letters.

4.1. Isotropy

We begin our methodology by proving that MARBERT's embedding space lacks uniformity. We follow (Mu et al., 2017) (Rajaee and Pilehvar, 2021a) in quantifying isotropy using the metric in 1, and the partition function 2 by (Arora et al., 2016):

$$I(W) = \frac{min_{u \in U} F(u)}{max_{u \in U} F(u)} \tag{1}$$

$$F(u) = \sum_{i=1}^{N} e^{u^T w_i} \tag{2}$$

where:

- U: is the set of eigenvectors of the embedding matrix W
- u: is the unit vector
- $F(u)$: is the partition function
- w_i: is the contextual word representation of the ith word in embedding matrix W where $W \in \mathrm{R}^{N \times D}$ with N being the size of the vocabulary and D the size of the embedding

As cited by (Rajaee and Pilehvar, 2021a), (Arora et al., 2016) proved that using a constant for isotropic embedding spaces, $F(u)$ can be approximated. Thereby $I(w)$ serves as an approximation for isotropy, where the closer the value it yields is to one the more isotropic the embedding space is.

[2] https://sites.google.com/view/arabichate2022/home

5. System Description

An essential part of our methodology is proving firstly that MARBERT's embedding space is in need for a better distribution. To quantify this, we calculate MARBERT's isotropy by extracting the training samples corresponding representations from MARBERT and apply equations 1 & 2. Furthermore, we visualise the embedding space to display non-uniformity of representations in Figure 1.

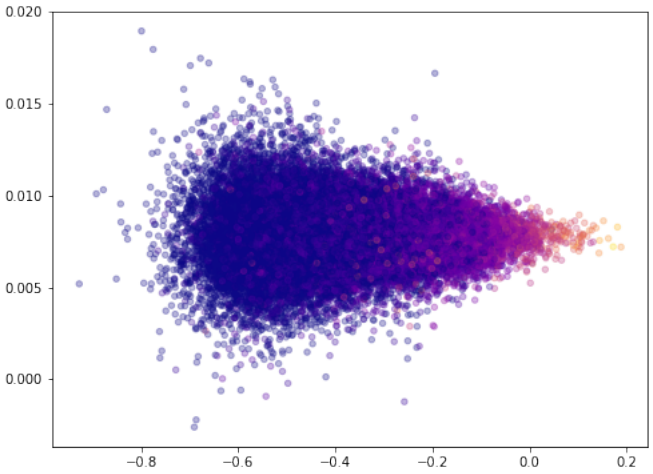

Figure 2: Visual of MARBERT's Cone-Shaped Embedding Space

We find MARBERT's isotropy value to be **2.88e-08**, which means that MARBERT has an extremely anisotropic embedding space, meaning that representations in MARBERT are not uniformly scattered around the origin. Figure 2 is an illustration of MARBERT's embedding space and shows that the model exhibits the representation degeneration problem. Darker colours in the figure indicate low frequency words while light colors are words with high frequency. When inspecting the figure we find the majority of low frequency words clustered together and shifting away from the origin. Moreover, the shape of the embedding takes the shape of a cone aligning with the depiction of an anisotropic space mentioned in 3.

5.1. Improving Isotropy

To refine MARBERT's isotropy, we use a cluster-based approach (Rajaee and Pilehvar, 2021a) which builds on top of (Mu et al., 2017) technique to improve isotropy in non-contextual word embeddings. Experiments in (Rajaee and Pilehvar, 2021a) illustrated the use of such method which improved classification performance and surpassed baseline values in pre-trained language models, particularly, BERT(Devlin et al., 2018).
The method is composed of the following steps:

1. cluster extracted features into k using K-means
2. zero-mean each cluster
3. remove a specified number of dominant directions D in each cluster. These dominant representations are the most occurring words per cluster

We set D according to (Devlin et al., 2018) approach.

$$D = \frac{d}{100} \tag{3}$$

where d is the dimension of a word embedding.
However for k we experiment with different ks and measure the isotropy for each k setting. Table 1 shows isotropy values after using cluster-based approach to enhance representations. Isotropy values show there is a significant increase starting with k=1 compared to the original baseline value. Isotropy values then sustain a small increase as the number of k clusters increases.

Baseline	**2.88e-08**
k =1	0.9564
k =3	0.9685
k = 5	0.9728
k =7	0.9719
k = 10	0.9743
k = 20	0.9767

Table 1: Isotropy Values for Different k Values

We illustrate in Figure 4 the embedding space post cluster-based approach. As shown, the geometry of representations completely changed from being cone-shaped to clusters scattered around the origin.

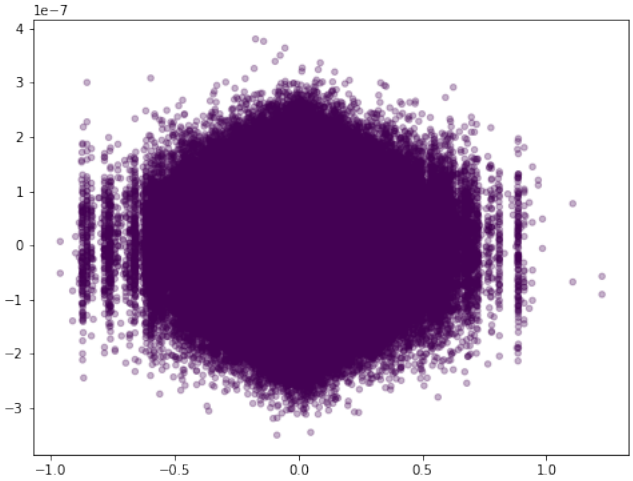

Figure 3: Representations post cluster-based processing

5.2. Training, Development & Testing

Next, We pass our isotropic representations to a Bidirectional Long-Short Term Memory (biLSTM) to be learned and perform classification.

5.2.1. Experimental Setup

We opt for using 10 clusters, and eight dominant directions to be removed per each according to the formula in 3. Naturally, we would not want to remove a larger number of directions not to lose linguistic features of the dataset.

Our biLSTM model is made up of two hidden layers each of size 102 and input sequence length of 768. We use Adam optimizer with decoupled weight regularization, a learning rate of 3e-5 and cross entropy loss. The development data undergoes the same preprocessing as the training data, meaning, we apply cluster based approach on development data as well with the same settings used with training samples. We validate our results using development data every epoch for 10 epochs of training and save the best performing checkpoint according to macro-averaged f1 score.

Additionally, we extract representations from MARBERT without any isotropic processing and compare development scores between isotropic representations and default ones.

6. Results & Discussion

Rep.	accuracy	precision	recall	f1
Isotropic	**81.2%**	71.5%	**71.5%**	**79.1%**
Default	81.2%	**74.9%**	61.5%	77.2%

Table 2: Development scores per type of Representations used

Table 2 shows results for the development data achieved by using isotropic representations in contrast to MARBERT's raw representations. Using better representations boosted f1 score by 1.9%.

Given these scores, we submit the better performing system of the two for the testing phase to achieve an accuracy of 76.5%, and 74.5% for f1. Additionally, our system scores 76.5% and 74.2% for precision and recall.

The confusion matrix of our better performing system (isotropic system) on the development set in figure 4 shows that it managed to predict 766 out of 898 non-offensive tweets correctly while predicting 271 offensive tweets correctly out of 368.

We employ different settings to our clustering approach in-order to enhance performance.

6.1. Additional Settings

We tried to anchor the number of dominant directions against distant values of k in order to check if the number of clusters in an embedding space can have an impact on the performance. We choose to compare between k = 2, 10, & 20 and we report below their respective validation results .

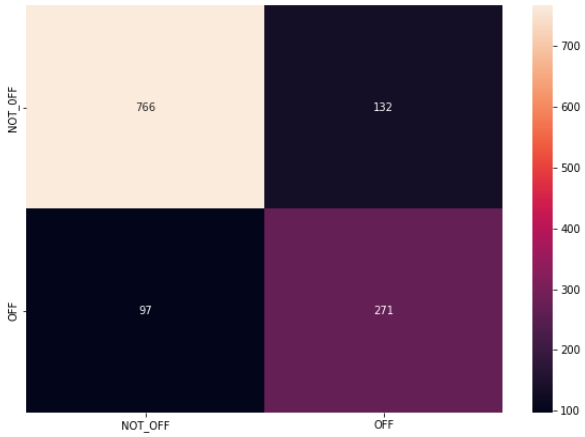

Figure 4:

K	Accuracy	F1
k =2	81.1%	71.1%
k =10	81.2%	79.1%
k = 20	81.0%	78.9%

Table 3: Development Results using k=2, 10 & 20

From table 3 we observe that scores are very close, and using all ks gives better performance than the default anisotropic representations. This means that increasing or lowering k may not impact performance significantly as long as the use of those k-values showcase isotropic properties in representations. Furthermore, we experimented with removing a larger number of dominant directions such as 16, and 30 directions, to find that the performance stays relatively the same.

This suggests one of two things, the first being is that we may have removed too many directions, thus some essential semantics were perished, or, given the nature of twitter data as we can see in figure 2, the majority of words are of low frequency (dark coloured points), hence, there is a limited number of dominant directions to be removed.

7. Conclusion

In this shared task we contributed to sub-task A, and in order to classify offensive and non-offensive tweets we aimed improve the geometric properties of MARBERT's embedding space, where we used a cluster-based approach to group representations into a number of clusters, removed a number of dominant directions per cluster which improved isotropy from being originally **2.88e-08** to **0.9743**. Moreover, we experiment the use of isotropic representations in contrast to

MARBERT's anisotropic embeddings by feeding each to a Bidirectional LSTM to find that isotropic representations achieve better scores. Our system for task A achieves an accuracy score of **81.2%** and f1 score of **79.1%** for development data and **76.5%** and **74.2%** for accuracy and f1 scores using test data respectively surpassing the baseline and outperforming MARBERT's default embeddings. Our code for the task can be found on github.[3]

8. Bibliographical References

Abdul-Mageed, M., Elmadany, A., and Nagoudi, E. M. B. (2020). Arbert & marbert: deep bidirectional transformers for arabic. *arXiv preprint arXiv:2101.01785*.

Abuzayed, A. and Elsayed, T. (2020). Quick and simple approach for detecting hate speech in arabic tweets. In *Proceedings of the 4th workshop on open-source Arabic Corpora and processing tools, with a shared task on offensive language detection*, pages 109–114.

Arora, S., Li, Y., Liang, Y., Ma, T., and Risteski, A. (2016). A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399.

Church, K. W. (2017). Word2vec. *Natural Language Engineering*, 23(1):155–162.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Gao, J., He, D., Tan, X., Qin, T., Wang, L., and Liu, T.-Y. (2019). Representation degeneration problem in training natural language generation models. *arXiv preprint arXiv:1907.12009*.

Huang, L., Yang, D., Lang, B., and Deng, J. (2018). Decorrelated batch normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 791–800.

Husain, F. (2020). Osact4 shared task on offensive language detection: Intensive preprocessing-based approach. *arXiv preprint arXiv:2005.07297*.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., and Mikolov, T. (2016). Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Liang, Y., Cao, R., Zheng, J., Ren, J., and Gao, L. (2021). Learning to remove: Towards isotropic pre-trained bert embedding. In *International Conference on Artificial Neural Networks*, pages 448–459. Springer.

Mu, J., Bhat, S., and Viswanath, P. (2017). All-but-the-top: Simple and effective post-processing for word representations. *arXiv preprint arXiv:1702.01417*.

Mubarak, H., Hassan, S., and Chowdhury, S. A. (2022). Emojis as anchors to detect arabic offensive language and hate speech. *arXiv preprint arXiv:2201.06723*.

Rajaee, S. and Pilehvar, M. T. (2021a). A cluster-based approach for improving isotropy in contextual embedding space. *arXiv preprint arXiv:2106.01183*.

Rajaee, S. and Pilehvar, M. T. (2021b). An isotropy analysis in the multilingual bert embedding space. *arXiv preprint arXiv:2110.04504*.

Rajaee, S. and Sheikhi, M.). Isotropic contextual word representation in bert.

Saeed, H. H., Calders, T., and Kamiran, F. (2020). Osact4 shared tasks: Ensembled stacked classification for offensive and hate speech in arabic tweets. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 71–75.

Soliman, A. B., Eissa, K., and El-Beltagy, S. R. (2017). Aravec: A set of arabic word embedding models for use in arabic nlp. *Procedia Computer Science*, 117:256–265.

Wang, L., Huang, J., Huang, K., Hu, Z., Wang, G., and Gu, Q. (2019). Improving neural language generation with spectrum control. In *International Conference on Learning Representations*.

[3] `https://github.com/nehalelkaref/Towards-a-Better-Isotropy-to-Detect-Hatespeech.git`