Proceedings of the OSACT 2022 Workshop @LREC2022, pages 162–166
Marseille, 20 June 2022

Overview of OSACT5 Shared Task on Arabic Offensive Language and Hate Speech Detection

Hamdy Mubarak[1], Hend Al-Khalifa[2], AbdulMohsen Al-Thubaity[3]
[1] Qatar Computing Research Institute, HKBU, Doha, Qatar,
[2] Information Technology Department, King Saud University, Riyadh, KSA
[3] King Abdulaziz City for Science and Technology (KACST), Riyadh, KSA
humbarak@hbku.edu.qa, hendk@ksu.edu.sa, aalthubaity@kacst.edu.sa

Abstract

This paper provides an overview of the shard task on detecting offensive language, hate speech, and fine-grained hate speech at the fifth workshop on Open-Source Arabic Corpora and Processing Tools (OSACT5). The shared task comprised of three subtasks; Subtask A, involving the detection of offensive language, which contains socially unacceptable or impolite content including any kind of explicit or implicit insults or attacks against individuals or groups; Subtask B, involving the detection of hate speech, which contains offensive language targeting individuals or groups based on common characteristics such as race, religion, gender, etc.; and Subtask C, involving the detection of the fine-grained type of hate speech which takes one value from the following types: **(i)** race/ethnicity/nationality, **(ii)** religion/belief, **(iii)** ideology, **(iv)** disability/disease, **(v)** social class, and **(vi)** gender. In total, 40 teams signed up to participate in Subtask A, and 17 of them submitted test runs. For Subtask B, 26 teams signed up to participate and 12 of them submitted runs. And for Subtask C, 23 teams signed up to participate and 10 of them submitted runs. 10 teams submitted papers describing their participation in one subtask or more, and 8 papers were accepted. We present and analyze all submissions in this paper.

Keywords: OSACT, Arabic, Offensive Language, Hate Speech, Fine-Grained Hate Speech, Shared Task, CodaLab

1. Introduction

Disclaimer: Due to the nature of this work, some examples contain offensive language and/or hate speech. This does not reflect authors' opinions by any mean. Our aim is to detect and prevent such harmful content from spreading.

Detection of offensive language and hate speech is very important for content moderation, online safety, etc. Studies show that the presence of hate speech may be connected to hate crimes (Watch, 2014). In recent years, there has been a large amount of research on detecting offensive language and hate speech in the NLP and computational social sciences communities. Many shared tasks were created for this purpose such as OffensEval 2020 (Zampieri et al., 2020) to detect offensive language for five languages, and OSACT4 (Mubarak et al., 2020a) to detect offensive language and hate speech for Arabic.

OSACT5 shared task can be considered as an extension of OSACT4, where the target is to identify the fine-grained type of the hate speech in addition to detecting offensive language and hate speech on Arabic social media using a new dataset.

We considered any kind of socially unacceptable or impolite content as offensive language. This includes vulgar, swear words, and any kind of explicit or implicit insults or attacks against individuals or groups.

Hate speech contains offensive language targeting individuals or groups based on common characteristics such as Race (including also ethnicity and nationality),[1] Religion (including belief), Ideology (ex: political or sport affiliation), Disability (including diseases), Social Class, and Gender.[2]

The shared task has three subtasks. Subtask A involves the detection of offensive language, and Subtask B is concerned with detecting hate speech. Subtask C is concerned with detecting the hate speech type.

2. Dataset

We used the data set described in (Mubarak et al., 2022) which contains 12,698 tweets collected using emojis that commonly appear in offensive communications. These emojis are extracted from existing datasets of offensive tweets, namely (Zampieri et al., 2020) and (Chowdhury et al., 2020). Authors showed that using emojis is more efficient than keywords (ex, as in (Mubarak et al., 2017)) or patterns (as in (Mubarak et al., 2020b)) and this method can be applied to other languages to collect a large percentage of offensive and hate tweets regardless of their topics, dialects, or genres. Tweets were extracted from 4.4M Arabic tweets collected between June 2016 and November 2017 having one or more emojis from a predefined list.

Tweets were labeled using two jobs on Appen crowdsourcing platform with the following quality settings: 3 judgements per tweet, 200 test questions, and 80% threshold to pass test questions. Inter-Annotator Agreement agreement was 0.82 (Cohen's kappa value). In the first annotation job (Job1), annotators classified tweets into Offensive (OFF) or Clean (CLN). In Job2, offensive tweets obtained from Job1 were classified into one of the

[1]We merged close types to ease the task.

[2]Other hate speech types did not exist in the Arabic dataset.

fine-grained hate speech types. Examples and statistics are shown in Table 1.

The subtasks used the same splits as in (Mubarak et al., 2022) for training (70% of all tweets), development (10%), and testing (20%). For Subtask A (offensiveness detection), the labels are: OFF or NOT_OFF, and for Subtask B (hate speech detection), the labels are: HS or NOT_HS. For Subtask C (hate speech type), the labels are: HS1 (Race), HS2 (Religion), HS3 (Ideology), HS4 (Disability), HS5 (Social Class), and HS6 (Gender) in addition to NOT_HS.

Simple preprocessing steps were applied to tweets to replace user mentions with @USER, URLs with "URL", and empty lines with <LF>.

3. Task Settings and Evaluation

Given the strong imbalance in class distributions in all Subtasks, we used the macro-averaged F1-score ($\mathcal{F}$) as the official evaluation measure. Macro-averaging gives equal importance to all classes regardless of their size. We also used Precision ($\mathcal{P}$) and Recall ($\mathcal{R}$) on the positive class (offensive or hate speech tweets) in addition to the overall Accuracy ($\mathcal{A}$) as secondary evaluation measures.

Subtasks were hosted on CodaLab platform at the following competition links:
Subtask A: `https://codalab.lisn.upsaclay.fr/competitions/2324`
Subtask B: `https://codalab.lisn.upsaclay.fr/competitions/2332`
Subtask C: `https://codalab.lisn.upsaclay.fr/competitions/2334`

We allowed teams to submit up to 10 runs on the test set, and we asked them to specify two submissions as their official runs (primary/first and secondary/second submissions). If they didn't specify their official runs, the latest were considered as official. Teams had the freedom to describe the differences between these runs in their papers which gives the chance to examine the effectiveness of different approaches and setups.
The official score for all subtasks was the macro-average F1 ($\mathcal{F}$) of the first submission.

The shared task attracted a large number of participants. In all, 40, 26 and 23 teams signed up to Subtasks A, B and C respectively. From them, 17, 12 and 10 teams submitted test runs to Subtasks A, B and C in order. Of those teams, 10 submitted system description papers and 8 papers were accepted. Table 2 lists information about the accepted papers, teams and affiliations.

We received 142 submissions for Subtask A including 22 failed ones (due to incorrect format). For Subtask B, we received 70 submissions including 3 failed ones. And for Subtask C, we received 59 submissions including 4 failed ones. Competitions were open from March 1st, 2022 until March 30th, 2022. The test sets were available starting from March 26th, 2022.

4. Results and Methods

The highest F1 score for Subtask A was 0.852 (Accuracy = 0.867, Precision = 0.856, and Recall = 0.848) achieved by **GOF** team (Mostafa et al., 2022). For Subtask B, the highest F1 was 0.831 (Accuracy = 0.941, Precision = 0.869, and Recall = 0.801) achieved by **iCompass** team (Ben Nessir et al., 2022). And for Subtask C, the highest F1 was 0.528 (Accuracy = 0.919, Precision = 0.548, and Recall = 0.531) achieved also by **iCompass** team (Ben Nessir et al., 2022).

Most teams performed basic to extensive data preprocessing, which typically involved character normalization, removal of punctuation, diacritics, repeated letters, and non-Arabic tokens. As for learning methods, the teams used different fine-tuned transformer versions, such as mT5, AraBERT, ARBERT, MARBERT, AraElectra, QARiB, Albert-Arabic, AraGPT2, mBert, and XLMRoberta.

The highest ranking submissions used an ensemble of different transformers. Table 3 briefly lists the preprocessing and learning methods used by different teams. Tables 4, 5, and 6 list the results of all the teams for Subtasks A, B, and C in order ranked by F1-measure ($\mathcal{F}$).

5. Conclusion

This paper presented an overview of the OSACT5 shared task on offensive language and hate speech detection in the Arabic Twitter sphere. The shared task consists of three subtasks: A, B, and C. The most successful systems in the shared task performed Arabic specific preprocessing, with the winning system for hate speech detection (subtask A) performing an ensemble of different machine learning approaches, while the the winning system for offensive language detection (subtask B) used a multi-task of different pre-trained language models, and finally, the winning system for the detection of the fine-grained type of hate speech detection (subtask C) used task specific layers that were fine-tuned with Quasi-recurrent neural networks (QRNN).

6. References

AlKhamissi, B. and Diab, M. (2022). Meta ai at arabic hate speech 2022: Multitask learning with self-correction for hate speech classification. *OSACT*, 5.

Alzu'bi, S., Ferreira, T. C., Pavanelli, L., and Al-Badrashiny, M. (2022). aixplain at arabic hate speech 2022: An ensemble based approach to detecting offensive tweets. *OSACT*, 5.

Ben Nessir, M. A., Rhouma, M., Haddad, H., and Fourati, C. (2022). icompass at arabic hate speech 2022: Detect hate speech using qrnn and transformers. *OSACT*, 5.

Chowdhury, S. A., Mubarak, H., Abdelali, A., Jung, S.-g., Jansen, B. J., and Salminen, J. (2020). A multi-platform arabic news comment dataset for offensive language detection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6203–6212.

de Paula, A. F. M., Rosso, P., Bensalem, I., and Zaghouani, W. (2022). Upv at the arabic hate speech 2022 shared task: Offensive language and hate speech detection using transformers and ensemble models. *OSACT*, 5.

Table 1: Statistics and examples from the annotated corpus

Class/Subclass	#	%	Example
Clean (CLN or NOT_OFF)	8,235	65	لن تحصل على غدٍ افضل مادمت تفكر بالامس You won't have a better tomorrow as long as you think about yesterday.
Offensive (OFF)	4,463	35	يلعن ابوك على هالسؤال. عساه ينقرض الكريه May God curse your father for this question! I hope this fool will die out!
Hate Speech	1,339	11	(Note: 30% of Offensive tweets are labeled as Hate Speech)
- Gender	641	48	بنات اليوم قليلات أدب. والله ما نوصل لعهر بعض الرجال Girls today are impolite. I swear to God, we don't reach for some men immorality.
- Race	366	27	شعبكم متخلف. الله ياخذك إنتي والفلبين People of your country are musty. May God take (kill) you and the Philippines.
- Ideology	190	14	ناديك وضيع لا شك في ذلك. حزبك لا يقدر إلا على النباح You club is vile, no doubt about that. Your party cannot do anything except barking.
- Social Class	101	8	دامك مقيم انكتم وخل اهل الأرض الاصليين يتكلمون. ابلع يا سباك! As you are a resident, shut up and let original citizens speak. Swallow, plumber!
- Religion	38	3	إنتوا بتعملوا ف ديك أبونا كده ليه هو إحنا كفرة ولايهود Why are you doing this to us? Are we disbelievers or Jews?
- Disability	3	0	ذا القزم طلعت جايزتن له بس ماعرف يعبر This dwarf got two prizes, but he does not know how to express.

Team	Affiliation	Subtasks
aiXplain (Alzu'bi et al., 2022)	aiXplain Inc, USA	A
iCompass (Ben Nessir et al., 2022)	iCompass, Tunisia	A, B, C
AlexU-AIC (Shapiro et al., 2022)	Alexandria University, Egypt	A, B, C
CHILLAX (Makram et al., 2022)	Helwan University, Egypt	A, B
GOF (Mostafa et al., 2022)	Helwan University, Egypt	A
GUCT (Elkaref and Abu-Elkheir, 2022)	German University in Cairo, Egypt	A
Meta-AI (AlKhamissi and Diab, 2022)	Meta, USA	A, B, C
UPV (de Paula et al., 2022)	Universitat Politecnica de Valencia, Spain	A, B, C

Table 2: List of participating teams in Subtasks A, B, and C (alphabetical order)

Elkaref, N. and Abu-Elkheir, M. (2022). Guct at arabic hate speech 2022: Towards a better isotropy for hate speech detection. *OSACT*, 5.

Makram, K. H., Nessim, K. G., Abd-Almalak, M. E., Roshdy, S. Z., Salem, S. H., Thabet, F. F., and Mohamed, E. H. (2022). Chillax - at arabic hate speech 2022: A hybrid machine learning and transformers based model to detect arabic offensive and hate speech. *OSACT*, 5.

Mostafa, A., Mohamed, O., and Ashraf, A. (2022). Gof at arabic hate speech 2022: Breaking the loss function convention for data-imbalanced arabic offensive text detection. *OSACT*, 5.

Mubarak, H., Darwish, K., and Magdy, W. (2017). Abusive language detection on arabic social media. In *Proceedings of the first workshop on abusive language online*, pages 52–56.

Mubarak, H., Darwish, K., Magdy, W., Elsayed, T., and Al-Khalifa, H. (2020a). Overview of osact4 arabic offensive language detection shared task. In *Proceedings of the 4th Workshop on open-source arabic corpora and processing tools, with a shared task on offensive language detection*, pages 48–52.

Mubarak, H., Rashed, A., Darwish, K., Samih, Y., and Abdelali, A. (2020b). Arabic offensive language on twitter: Analysis and experiments. *arXiv preprint arXiv:2004.02192*.

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*.

Shapiro, A., Khalafallah, A., and Torki, M. (2022). Alexu-aic at arabic hate speech 2022: Contrast to classify. *OSACT*, 5.

Watch, H. S. (2014). Hate speech watch. hate crimes: Consequences of hate speech. In *http://www.nohatespeechmovement.org/hate-speechwatch/focus/ consequences-of-hate-speech*.

Zampieri, M., Nakov, P., Rosenthal, S., Atanasova, P., Karadzhov, G., Mubarak, H., Derczynski, L., Pitenis, Z., and Çöltekin, Ç. (2020). Semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020). *arXiv preprint arXiv:2006.07235*.

Team	Preprocessing	Methods
aiXplain (Alzu'bi et al., 2022)	For the textual part of the tweet, they apply the following transformations sequentially on each tweet: 1. Remove URLs and mentions, 2. Remove diacritics and tatweel, 3. Remove punctuation. For the Emojis part, they translated emojis in a tweet to Arabic using (Junczys-Dowmunt et al., 2018) English to Arabic model. For some emojis, they inferred their intended meaning and provided their translation using the team's expertise in the Arabic language and the colloquial dialect used. Additionally, they extracted the relevant emojis from each tweet and used a classifier to predict their sentiment individually. They also used data augmentation (Semi-Supervised Learning and Contextual Augmentation based on Semantic Similarity).	Their system architecture involved feeding the predictions of an ensemble of classifiers combined with relative high-level features to a final meta-learner yielding a binary label of "OFF" to represent offensive or "NOT_OFF" to represent inoffensive speech. Each of the classifiers in the ensemble consist of a final linear layer the following pre-trained model as a backbone: AraBERTv0.2-Twitter-large - Mazajak 250M CBOW pre-trained embeddings - Character level N-gram + word level N-gram TF- IDF embeddings - MUSE. The predictions from the aforementioned models are then concatenated into a final vector.
iCompass (Ben Nessir et al., 2022)	1. Removing all non Arabic tokens, including ones like USER, URL, < LF >. Emojis were also removed. 2. Normalizing all the hashtags by simply decomposing them. 3. Removing white spaces.	Different pre-trained models were used in order to achieve the best results when fine-tuning it in a multi-task fashion (mT5, AraBERT, ARBERT, and MARBERT) and task specific layers that were fine-tuned with Quasi-recurrent neural networks (QRNN) for each down-stream subtask.
AlexU-AIC (Shapiro et al., 2022)	Arabic letters, punctuation and digit Normalization, Hashtag segmentation, diacritic and symbols removal and removal of repeated characters or emojis more than two times	AraBERT, MarBERT v1 and MarBERT v2 with multiple training paradigms such as: Classification Fine-tuning, Contrastive Learning and Multi-task Learning.
CHILLAX (Makram et al., 2022)	cleaning: all URLs and User mentions were removed. augmentation: generates new tweets from the minority classes using MARBERT Arabic model	MARBERT Arabic LM for features extraction and Logistic Regression and Random Forest for training.
GOF (Mostafa et al., 2022)	non-Arabic letters, punctuation marks, digits, Arabic diacritics and repeated characters removal and replacing URL, @USER, and Email with their Arabic translations (رابط، مستخدم، بريد)	seven language models: MARBERT(without emojis), AraBERT-Large-Twitter, QARiB, AraBERT-Base-Twitter, MARBERT, MARBERTV2, LightGBM(QARiB Embeddings) and ensemble learning approach : Ensemble(LightGBM+ MARBERT+MARBERTV2) ,Ensemble(AraBERT-B-T+ MARBERT+QARiB) and Ensemble(MARBERTV2+ MARBERT+QARiB)
GUCT (Elkaref and Abu-Elkheir, 2022)	replace any instances of Twitter mentions with "@USER" and URLs by "URL". diacritics and non-Arabic letters removal.	1. calculate MARBERT's isotropy. 2. refine MARBERT's isotropy. 3. pass refined isotropic representations to a Bidirectional Long-Short Term Memory (biLSTM) to be learned and perform classification.
Meta-AI (AlKhamissi and Diab, 2022)	user mentions are reduced to @USER, URLs are replaced with URL , and empty lines in original tweets are replaced with <LF>.	the input text is encoded using MARBERTv2 and is then passed to 3 task-specific classification heads. Each class specific head is made up of a multi-layered feed forward neural network with layer normalization.
UPV (de Paula et al., 2022)	No preprocessing	six different transformer versions: Arabert, AraElectra, Albert-Arabic, AraGPT2, mBert, and XLMRoberta. In addition, two ensemble methods were employed: Majority vote and Highest sum

Table 3: Methods used by different teams (alphabetical order)

	First Submission				Second Submission			
Team	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
GOF (Mostafa et al., 2022)	**0.867**	**0.856**	**0.848**	**0.852**	0.864	0.853	0.844	0.848
Meta AI (AlKhamissi and Diab, 2022)	0.860	0.846	0.843	0.845	0.852	0.839	0.834	0.836
aiXplain (Alzu'bi et al., 2022)	0.858	0.845	0.840	0.843	0.864	0.852	0.847	0.849
AlexU-AIC (Shapiro et al., 2022)	0.856	0.842	0.839	0.841				
iCompass (Ben Nessir et al., 2022)	0.854	0.841	0.837	0.839	-	-	-	-
UPV (de Paula et al., 2022)	0.837	0.821	0.818	0.819	0.841	0.824	0.831	0.827
CHILLAX (Makram et al., 2022)	0.803	0.784	0.779	0.781	0.740	0.716	0.723	0.719
GUCT (Elkaref and Abu-Elkheir, 2022)	0.765	0.742	0.750	0.745	-	-	-	-
BASELINE	0.651	0.325	0.500	0.394	-	-	-	-

Table 4: Subtask A results

	First Submission				Second Submission			
Team	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
iCompass (Ben Nessir et al., 2022)	**0.941**	0.869	**0.801**	**0.831**				
Meta-AI (AlKhamissi and Diab, 2022)	**0.941**	**0.870**	0.795	0.827	0.938	0.845	0.819	0.832
AlexU-AIC (Shapiro et al., 2022)	0.937	0.855	0.787	0.817				
CHILLAX (Makram et al., 2022)	0.891	0.728	0.809	0.759	0.869	0.694	0.792	0.727
UPV (de Paula et al., 2022)	0.925	0.845	0.711	0.757	0.932	0.858	0.751	0.792
BASELINE	0.893	0.447	0.500	0.472	-	-	-	-

Table 5: Subtask B results

	First Submission				Second Submission			
Team	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
iCompass (Ben Nessir et al., 2022)	0.919	0.548	**0.531**	**0.528**				
Meta-AI (AlKhamissi and Diab, 2022)	**0.926**	**0.551**	0.508	0.519				
AlexU-AIC (Shapiro et al., 2022)	0.923	0.490	0.470	0.476				
UPV (de Paula et al., 2022)	0.920	0.543	0.369	0.423	0.917	0.382	0.294	0.325
BASELINE	0.893	0.128	0.143	0.135	-	-	-	-

Table 6: Subtask C results