ASAD: Arabic Social media Analytics and unDerstanding

Sabit Hassan, Hamdy Mubarak, Ahmed Abdelali, Kareem Darwish
Qatar Computing Research Institute
Hamad bin Khalifa University
Doha, Qatar
{sahassan2,hmubarak,aabdelali,kdarwish}@hbku.edu.qa

Abstract

This system demonstration paper describes the Arabic Social media Analysis and unDerstanding (ASAD) toolkit, which is a suite of seven individual modules that allows users to determine dialects, sentiment, news category, offensiveness, hate speech, adult content, and spam in Arabic tweets[1]. The suite is made available through a web API and a web interface where users can enter text or upload files.

1 Introduction

Since Arabic is spoken across a vast region, the Arabic Twittersphere presents a valuable scope into social and linguistic phenomena, such as the multitude of dialects being used across different regions. The Arabic Social Media and unDerstanding (ASAD) suite [2], which we present herein, offers valuable tools for exploring such phenomena and for the automated processing of Arabic social media texts. Specifically, ASAD offers dialect identification, sentiment analysis, news category detection, offensive language detection, including hate speech and vulgar language, and spam detection. These tools are valuable for many downstream NLP application. For example, dialect identification can help improve author profiling and machine translation (Abdelali et al., 2020). Sentiment analysis can aid in quantifying public opinions (Abu Farha and Magdy, 2019). Detecting news categories can aid in content analysis. Further, offensive language and spam detection can help identify potentially malicious content on social media. Although there has been a growing interest in analyzing Arabic social media, there is a deficiency in publicly available tools or such tools are not integrated into one framework or toolkit. For example, we are not aware of any publicly available systems for offensive language, hate speech, adult content, or spam. Similarly, ADIDA (Obeid et al., 2019) and CAMeL (Obeid et al., 2020) dialect identification systems were not trained with Twitter data. Thus, ASAD fills an important gap in the Arabic social media analysis space. For ease of use, we make ASAD available via an i) online interface where users can enter text or upload files, and ii) web APIs that accept POST requests, making ASAD accessible from any programming language.

During the development of ASAD, we weighed different trade-off between effectiveness and efficiency to achieve competitive results at low computational costs. Thus, ASAD utilizes Support Vector Machine (SVM) classification for six out of the seven modules. As we show later, with the exception of dialect identification, we achieve results that are comparable or slightly lower than deep neural network models (DNN), namely fine-tuned BERT, while being significantly more efficient with no need for GPUs. Due to a larger difference in performance, we deploy a fine-tuned BERT model for dialect identification only. We hope that ASAD will aid researchers, analysts, and system integrators in incorporating Arabic social media analytics and understanding into their models and applications. We also hope that ASAD will motivate researchers to build similar suites for other languages.

2 Related Work

Analysis of Arabic social media has gained much recent interest. Offensive language and hate speech detection have yielded datasets, shared tasks (Mubarak et al., 2020b; Zampieri et al., 2020), and strong systems based on machine learning and contextual embedding models (Hassan et al., 2020a,b). Sentiment analysis is a well addressed problem yielding datasets (Elmadany et al., 2018)

[1] We will add more functionalities in the future.
[2] Demonstration: https://www.youtube.com/watch?v=Boe_JYWK7cM

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, pages 113–118
April 19 - 23, 2021.

and systems based on and deep learning techniques (Abu Farha and Magdy, 2019) among others. Fine-tuned BERT models have been used for identifying categories of news posts on social media (Chowdhury et al., 2020). Adult content and spam detection have been relatively less explored with the focus mainly on creating resources (Alshehri et al., 2018; Al Twairesh et al., 2016; Mubarak et al., 2017, 2021, 2020a). Dialect ID has been the focus of the MADAR project (Bouamor et al., 2019) and other works (Abdelali et al., 2020; Abdul-Mageed et al., 2020; Zaidan and Callison-Burch, 2011).

Despite the abundance of literature in the aforementioned topics, there has been very little effort toward making tools available for public use. Most of the tools available in Arabic NLP tasks concentrate on NLP tasks such as segmentation, parsing, lemmatization, and POS tagging (Pasha et al., 2014; Abdelali et al., 2016; Darwish and Mubarak, 2016; Darwish et al., 2014). Along with text processing tools, CAMeL Tools (Obeid et al., 2020) allows sentiment analysis and dialect ID via a Python package. ADIDA (Obeid et al., 2019) is a web interface for dialect ID. The dialect ID systems of CAMeL Tools and ADIDA are based a parallel corpus of 25 Arabic city dialects in the travel domain.

3 Datasets

Dialect ID: We use the QADI dataset containing dialectal tweets from 18 countries (Abdelali et al., 2020). The training set contains 540K tweets automatically tagged for dialect and the test set contains 3.3K manually annotated tweets by native speakers from the 18 countries.

Sentiment Analysis: We use the ArSAS dataset (Elmadany et al., 2018) that contains 21K tweets that are labeled as Positive, Negative, Mixed or Neutral. We merge the Mixed and Neutral classes together (resulting in three classes) and split the data into 80/20 training and test splits.

News Categorization We use an in-house annotated dataset consisting of 30K news items from Aljazeera channel[3]. 80% of the data are used for training and 20% are used for testing. These news are manually annotated for different categories, namely: politics, economy, sports, culture-art, etc.

Offensive Language Detection: We use data of OffensEval 2020 shared task (Zampieri et al., 2020). The data consists of 8K tweets for training and 2K tweets tweets for testing that were manually annotated with whether they are offensive or not.

Hate Speech Detection: There are limited publicly available data for Arabic hate speech detection (Mubarak et al., 2020b). We use a publicly available dataset[4] that consists of tweet IDs annotated for whether they contain hate speech or not. Ignoring tweets that were not available at download time, we end up with 6.9K tweets.[5] We use 80% of the data for training and 20% for testing.

Adult Content Detection: We use the dataset presented in Mubarak et al. (2021). The data contains 50K tweets split into 80% for training and 20% for testing. Around 6K tweets (12% of all tweets) are manually verified to contain adult content. The rest are random tweets that are assumed not contain adult material since the percentage of adult content in tweets is very small.

Spam Detection: We use the dataset presented in Mubarak et al. (2020a). The dataset contains 9.8K tweets from 80 spam accounts (manually verified) that post spam tweets, along with 86K random tweets for training. The test set contains 2.7K tweets from 20 spam accounts (manually verified) that post spam tweets along with 25.6K random tweets. The assumption is that tweets from spam accounts are spam and that the vast majority of random tweets are not spam, because the percentage of spam is very small.

4 Classification Models

Some state-of-the-art (SOTA) techniques use complex models, typically DNN models, to achieve the best results. For ASAD, we want to have models that are small in size and easy to deploy while providing good results. To this end, we compare performances of fine-tuned BERT models and SVMs with character n-gram vectors weighted by term frequency-inverse document frequency (tf-idf) as features. As we show, the SVM models we employ are competitive with SOTA DNN models for majority of the modules of ASAD. The range of n-gram can influence the size of models and their performance. For each component in our suite, we experimented with different ranges of n-gram and calculated model size along with respective performance. Table 1 illustrates this study for offensive

[3] www.aljazeera.net

[4] https://github.com/raghadsh/Arabic-Hate-speech

[5] We plan to merge other datasets in future.

Classifier	Features	Size (classifier + vectorizer)	Acc%	P	R	F1
SVM	W[1-3]	30.7 MB	86.6	78.9	82.6	80.5
SVM	**C[1-3]**	**3.9 MB**	**91.2**	**88.8**	**82.4**	**85.0**
SVM	C[2-4]	14.5 MB	91.6	89.2	83.4	85.8
SVM	C[2-5]	37.5 MB	92.0	89.1	85.1	86.9
SVM	C[2-6]	73.4 MB	91.8	87.6	86.4	87.0
SVM	C[2-7]	120.5 MB	91.8	86.9	88.1	87.4

Table 1: Comparison of size vs performance on offensiveness detection. Ideal setting is bolded.

Module	Classes	Classifier	Features	Acc%	P	R	F1	BERT F1
Dialect ID	18	SVM	C[2-4]	54.5	60.9	54.6	54.1	**60.6**
Sentiment	3	SVM	C[1-3]	75.5	74.6	73.2	73.7	**75.8**
News Category	16	SVM	C[2-4]	84.2	57.3	54.1	54.8	**55.9**
Offensiveness	2	SVM	C[1-3]	91.2	88.8	82.4	85.0	**86.6**
Hate Speech	2	SVM	C[2-4]	79.1	74.4	76.2	**75.2**	75.1
Adult Content	2	SVM	C[1-3]	95.4	91.9	85.79	**88.5**	88.1
Spam	2	SVM	C[1-3]	99.4	99.3	97.3	98.3	**98.9**

Table 2: Performance of different ASAD modules compared to fine-tuned BERT models.

language detection (C and W refer to character and word, [a-b] denotes n-gram ranging from a to b. P, R and F1 stand for macro-averaged precision, recall and F1 respectively). We can see that going from an n-gram range of C[1-3] to C[2-7] increases model size (classifier + vectorizer) from 3.9 MB to 120.5 MB while improving the F1 score by 2.4. Although C[2-7] is a better system, C[1-3] is more suitable for deployment due to its small size. Table 2 lists performance of SVMs version compared to using BERT. When comparing to BERT models, we fine-tuned AraBERT(Antoun et al., 2020), a BERT-based model, pre-trained on Arabic news articles and Arabic Wikipedia. We fine-tune AraBERT by adding a fully-connected dense layer followed by a softmax classifier, minimizing the binary cross-entropy loss function for the training data. We use the PyTorch[6] implementation by HuggingFace[7] as it provides pre-trained weights and vocabulary. Aside from dialect ID, SVM models either beat BERT models or are within 1-2% away. We suspect that the SVM models were competitive because they were trained on Twitter data as opposed to BERT, which is trained on more formal text. For dialect ID, we opt to use the fine-tuned AraBERT model because it outperforms SVMs by a larger margin of 6.5%.

5 Interface

Design The ASAD web interface is available at: http://asad.qcri.org/
The user can select any of the modules from the tabs and test the performance on random samples and classify them to easily understand the different modules. The user can type a text to be classified. The classification results appear in a table so that earlier results can be referred to. We recognize that users may want to classify many tweets in one go without having to type them one at a time. To allow this, the users can upload a text file. Each line is classified by our system and users can download a file that contains predicted class and class probabilities. To prevent excessive usage, we limit allowed files to have at most 100 lines. We also use Google reCaptcha V2[8] to prevent bots from abusing our file upload system. Figure 1 shows the common layout for all components except for Dialect ID. For Dialect ID, we use a map to visualize results. To this end, we provide a heatmap showing the distribution of probabilities for different dialects. This allows users to easily determine which part of the world the input text is likely to come from. Figure 2 illustrates layout for dialect ID. We also allow users to send feedback to us. This will help us improve ASAD in the future.

[6] https://pytorch.org/
[7] https://github.com/huggingface/transformers
[8] https://developers.google.com/recaptcha/

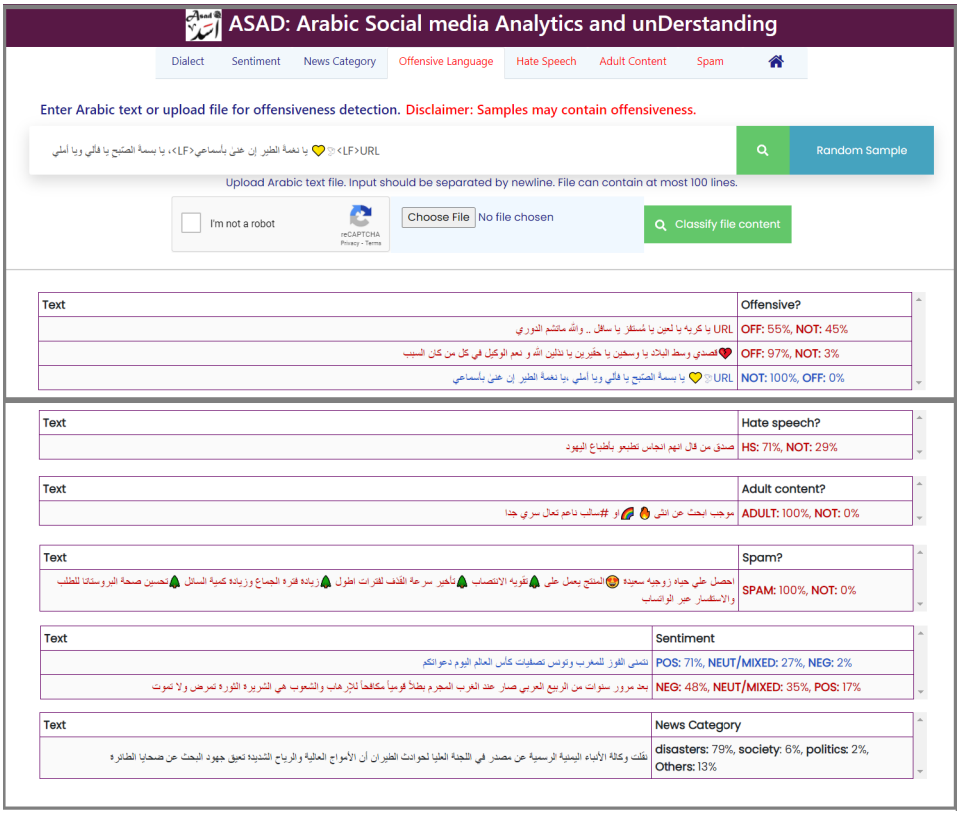

Figure 1: ASAD interface for Offensiveness module (top half) and outputs of other modules (lower half)

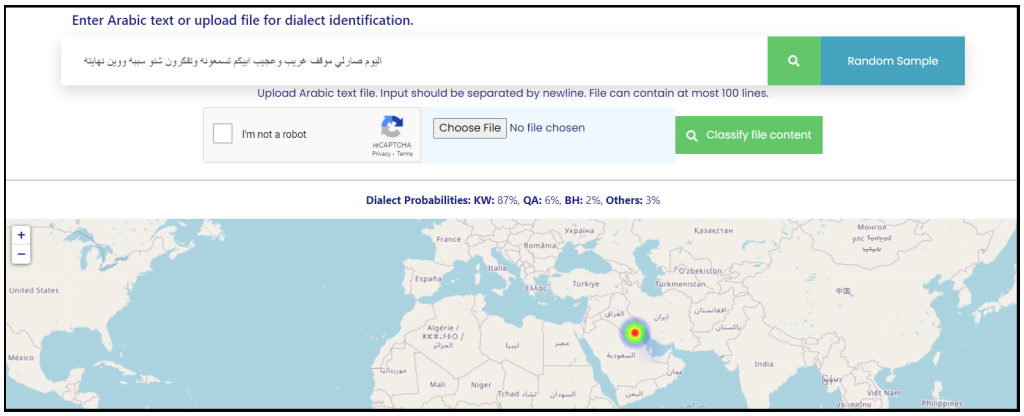

Figure 2: ASAD interface for Dialect ID module

Implementation We use Flask[9], a lightweight web application framework for backend development. Input from the user is first transformed into n-gram vectors using tf-idf vectorizer and then are passed to the classifiers (described in Section 4). The classifiers return predicted labels alongside probabilities of different classes. The class probabilities were calculated using Platt calibration (Platt, 1999). We use scikit-learn[10] to train all the SVM classifiers and vectorizers. We use javascript for functionality at frontend and for communication between the frontend and the backend. To display probabilities on a map for the dialect ID module, we use the heatmap layer plugin[11] with leaflet.js[12] and OpenStreetMap[13].

[9] https://palletsprojects.com/p/flask/

[10] https://scikit-learn.org/stable/index.html

[11] https://www.patrick-wied.at/static/heatmapjs/plugin-leaflet-layer.html

[12] https://leafletjs.com/

[13] https://www.openstreetmap.org/#map=8/25.322/51.197

Module	API URL	Body of request
Dialect ID	https://asad.qcri.org.com/dialect	KEY : text VALUE : arabic_text
Sentiment	https://asad.qcri.org/sentiment	
News Category	https://asad.qcri.org/news	
Offensiveness	https://asad.qcri.org/offensive	
Hate Speech	https://asad.qcri.org/hate_speech	
Adult Content	https://asad.qcri.org/adult	
Spam	https://asad.qcri.org/spam	

Table 3: API endpoints for ASAD

```
import requests
import json

text = "اليوم صارلي فوقف قريب وعجيب ابيكم تسمعونه وتفكرون شنو صيبه وبين نهايته"
url = "https://asad.qcri.org/dialect"
myobj = {'text': text}
request = requests.post(url, data = myobj)
ASAD_output = json.loads(request.text)
print ("==>", ASAD_output)

==> {'AE': '0.0', 'BH': '0.03', 'DZ': '0.0', 'EG': '0.0', 'IQ': '0.02', 'JO': '
0.0', 'KW': '0.87', 'LB': '0.0', 'LY': '0.0', 'MA': '0.0', 'OM': '0.0', 'PS': '
0.0', 'QA': '0.06', 'SA': '0.01', 'SD': '0.0', 'SY': '0.0', 'TN': '0.0', 'YE':
'0.0', 'prediction': 'KW'}
```

Figure 3: Example usage of ASAD Dialect ID API

Web API To facilitate using ASAD from different programming languages, we provide Web APIs via POST requests. Table 3 lists available API routes and Figure 3 illustrates example usage. Response from ASAD contains predicted class and class probabilities.

6 Conclusion

We presented ASAD, a system that can be used for analysis of tweets in multiple ways. Using one system, users can detect offensive language, hate speech, sentiment, news category, adult content, spam, and also identify dialects. For the ease of usage, our system can be both accessed via Web APIs and an online interface. In the future, we plan to release ASAD through the *pip* Python packaging tool.

References

Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. 2016. Farasa: A fast and furious segmenter for Arabic. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 11–16, San Diego, California. Association for Computational Linguistics.

Ahmed Abdelali, Hamdy Mubarak, Younes Samih, Sabit Hassan, and Kareem Darwish. 2020. Arabic dialect identification in the wild. *ArXiv*, abs/2005.06557.

Muhammad Abdul-Mageed, Chiyu Zhang, Houda Bouamor, and Nizar Habash. 2020. Nadi 2020: The first nuanced arabic dialect identification shared task. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 97–110.

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.

Nora Al Twairesh, Mawaheb Al Tuwaijri, Afnan Al Moammar, and Sarah Al Humoud. 2016. Arabic spam detection in twitter. In *The 2nd Workshop on Arabic Corpora and Processing Tools 2016 Theme: Social Media*, page 38.

Ali Alshehri, El Moatez Billah Nagoudi, Hassan Alhuzali, and Muhammad Abdul-Mageed. 2018. Think before your click: Data and models for adult content in arabic twitter.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. In *Proceedings of The 4th Workshop on Open-Source Arabic Corpora and Processing Tools*, pages 9–15.

Houda Bouamor, Sabit Hassan, and Nizar Habash. 2019. The MADAR shared task on Arabic fine-grained dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 199–207, Florence, Italy. Association for Computational Linguistics.

Shammur Absar Chowdhury, Ahmed Abdelali, Kareem Darwish, Jung Soon-Gyo, Joni Salminen, and Bernard J. Jansen. 2020. Improving Arabic text categorization using transformer training diversification.

In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 226–236, Barcelona, Spain (Online). Association for Computational Linguistics.

Kareem Darwish, Ahmed Abdelali, and Hamdy Mubarak. 2014. Using stem-templates to improve arabic pos and gender/number tagging. In *LREC*, pages 2926–2931. Citeseer.

Kareem Darwish and Hamdy Mubarak. 2016. Farasa: A new fast and accurate arabic word segmenter. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1070–1074.

AbdelRahim A. Elmadany, Hamdy Mubarak, and Walid Magdy. 2018. An arabic speech-act and sentiment corpus of tweets. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA). The 3rd Workshop on Open-Source Arabic Corpora and Processing Tools, OSACT3 ; Conference date: 08-05-2018.

Sabit Hassan, Younes Samih, Hamdy Mubarak, and Ahmed Abdelali. 2020a. ALT at SemEval-2020 task 12: Arabic and English offensive language identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1891–1897, Barcelona (online). International Committee for Computational Linguistics.

Sabit Hassan, Younes Samih, Hamdy Mubarak, Ahmed Abdelali, Ammar Rashed, and Shammur Absar Chowdhury. 2020b. ALT submission for OSACT shared task on offensive language detection. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 61–65, Marseille, France. European Language Resource Association.

Hamdy Mubarak, Ahmed Abdelali, Sabit Hassan, and Kareem Darwish. 2020a. Spam detection on arabic twitter. In *Social Informatics*, pages 237–251, Cham. Springer International Publishing.

Hamdy Mubarak, Kareem Darwish, and Walid Magdy. 2017. Abusive language detection on arabic social media. In *Proceedings of the First Workshop on Abusive Language Online*, pages 52–56.

Hamdy Mubarak, Kareem Darwish, Walid Magdy, Tamer Elsayed, and Hend Al-Khalifa. 2020b. 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*.

Hamdy Mubarak, Sabit Hassan, and Ahmed Abdelali. 2021. Adult content detection on arabic twitter: Analysis and experiments. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*.

Ossama Obeid, Mohammad Salameh, Houda Bouamor, and Nizar Habash. 2019. ADIDA: Automatic dialect identification for Arabic. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 6–11, Minneapolis, Minnesota. Association for Computational Linguistics.

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. CAMeL tools: An open source python toolkit for Arabic natural language processing. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France. European Language Resources Association.

Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 1094–1101, Reykjavik, Iceland. European Languages Resources Association (ELRA).

John C. Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press.

Omar F Zaidan and Chris Callison-Burch. 2011. The Arabic Online Commentary Dataset: an Annotated Dataset of Informal Arabic With High Dialectal Content. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 37–41.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *Proceedings of SemEval*.