AlexuNLP24 at AraFinNLP2024: Multi-Dialect Arabic Intent Detection with Contrastive Learning in Banking Domain

Hossam Elkordi, Ahmed Sakr, Marwan Torki, Nagwa ElMakky
Department of Computer and Systems Engineering
Alexandria University, Egypt
{es-hossam.elkordi2018,es-ahmedsakr20,mtorki,nagwamakky}@alexu.edu.eg

Abstract

Arabic banking intent detection represents a challenging problem across multiple dialects. It imposes generalization difficulties due to the scarcity of Arabic language and its dialects resources compared to English. We propose a methodology that leverages contrastive training to overcome this limitation. We also augmented the data with several dialects using a translation model. Our experiments demonstrate the ability of our approach in capturing linguistic nuances across different Arabic dialects as well as accurately differentiating between banking intents across diverse linguistic landscapes. This would enhance multi-dialect banking services in the Arab world with limited Arabic language resources. Using our proposed method we achieved second place on subtask 1 leaderboard of the AraFinNLP2024 shared task (Malaysha et al., 2024) with micro-F1 score of 0.8762 on the test split.

1 Introduction

With the increase in virtual assistants' popularity, it is crucial to maintain a high level of quality and efficiency while fulfilling the objectives and serving the users. This highlights the importance of spoken and natural language understanding systems in identifying a users' goal from their utterances and provide assistance to achieve this goal.

Intent detection is one of the core and essential tasks of natural language understanding (NLU) (Alshahrani et al., 2022). The goal is to extract syntactic and semantic information from utterances to be used in different downstream tasks such as conversation management (Hefny et al., 2020), question answering (Uva et al., 2020), etc.

This problem can further be classified based on the targeted semantic granularity. First, domain classification that identifies the topic the user is talking about such as banking, education, ...etc. The intent identification determines the specific outcome the user expects. At the lowest level comes the slot filling problem, where each word or span of words in the sentence is labelled based on the semantic information it provides relevant to the user's intent. Intent detection and slot filling are usually tightly coupled and can be embedded jointly together especially in conversation pipelines (Gangadharaiah and Narayanaswamy, 2019; Zhang et al., 2019).

Our research focuses on the banking domain in the Arab world where users' commands need to be accurately classified into a set of pre-defined supported services. This would benefit customer services and automatic query execution, especially after the increase in online banking usage in Arabic-speaking countries. While this increase is spanning across multiple countries, the need to handle different dialects has become more and more crucial.

While Arabic is recognized as the 4th most used language on the internet (Boudad et al., 2018; Guellil et al., 2021). Its dialects spanning, Classical Arabic (CA), Modern Standard Arabic (MSA), and Dialectal Arabic (DA), imposes difficulties in acquiring data at scale that is enough for building robust and efficient models that understand linguistic nuances while covering several domains.

In this paper, we present our submission to subtask-1 of the shared task AraFinNLP2024. Our main contribution lies in applying contrastive learning on a pre-trained BERT-based model to enable it to capture linguistic differences and similarities. This will improve the separation between different banking intents while being robust against dialectal nuances. Furthermore, we augmented the shared task's data through a secondary translation model, which will enable our model to understand dialects that are not available in the training split.

2 Related Work

In the deep learning era, different variants of CNNs and RNNs were heavily used in intent detection

Proceedings of The Second Arabic Natural Language Processing Conference, pages 415–421

task (Ravuri and Stolcke, 2016; Liu and Lane, 2016; Ali et al., 2020). In 2017, the Transformer architecture (Vaswani et al., 2017) has emerged and quickly become the baseline of most of NLP problems. Many studies used fine-tuned BERT (Devlin et al., 2019) and its variations for sequence classification tasks such as sentiment analysis (Araci, 2019) and intent detection (Casanueva et al., 2020).

To overcome challenges presented specifically by the Arabic language, multiple approaches are adapted. Back-translation was used to solve the dialectal imbalance problem in many tasks such as (Tahssin et al., 2020). (Duwairi and Abushaqra, 2021) tries to solve the data scarcity problem by augmenting the existing data with synonym-replaced sentences or negated sentences with their appropriate labels. Contrastive learning was used by (Shapiro et al., 2022) to overcome the problem of overfitting when finetuning large language models on small-sized datasets.

In the financial domain, Finbert (Liu et al., 2021) was developed by pre-training a BERT model on financial text corpus. It was then used in many tasks such as stock prediction (Chen, 2021). Efforts in the financial Arabic domain are still lacking. This is due to the lack of large annotated datasets. Hence many solutions applied in the English domain are still to be investigated and adapted for Arabic domain.

3 Dataset

We were provided with the ArBanking77 dataset (Jarrar et al., 2023) by the organizers of the shared task AraFinNLP2024 (Malaysha et al., 2024). This dataset is a translated version of the Banking77 dataset introduced in (Casanueva et al., 2020). The data includes MSA and Palestinian (PAL) dialects only in the train and validation splits. The test split, on the other hand, contains 11,721 samples with added Moroccan (MOR), Tunisian (TUN), and Saudi (SAU) dialects. To overcome this issue, we augmented our training data with these three dialects by translating the MSA part of the training data to these dialects using NLLB (No Language Left Behind) translation model (Costa-jussà et al., 2022) which is good in translating low-resource languages. Table 1 shows the distribution of the dialects in the train and the validation split. After adding the translated data the dialects are almost equally distributed.

The dataset consists of 77 banking intents that are not equally distributed. Both training and validation split have similar label distribution. Intents related to payments are the most dominant (above 2%) such as *card_payment_wrong_exchange_rate and transfer_not_received_by_recipient* classes. Most of the other classes range between 1-2% and some classes are very underrepresented (under 0.5%) such as *contactless_not_working and card_acceptance*. This imbalance can cause the model to be biased to dominant intents. To tackle this issue we weighted each class during the training of the classification model so that the error in each intent would have the same effect on the training process.

Split	MSA	PAL	MOR	TUN	SAU
O-Tr	10,733	10,821	0	0	0
O-Val	1,230	1,234	0	0	0
A-Tr	0	0	9,694	9,694	9,694
A-Val	0	0	1,039	1,039	1,039
Tr	10,733	10,821	9,694	9,694	9,694
Val	1,230	1,234	1,039	1,039	1,039

Table 1: Distribution of dialects in the train (Tr) and validation (Val) splits of the original (O) data and the augmented dataset (A) that we translated using NLLB.

4 System Overview

4.1 Backbone

After the emergence of the Transformer architecture (Vaswani et al., 2017) in 2017, it has quickly become the baseline of most of NLP problems. BERT model (Devlin et al., 2019) and its variations represent the state-of-the-art in the intent detection problem. Many studies (He et al., 2019; Casanueva et al., 2020; Abro et al., 2022) fine-tuned them on domain-specific datasets using different learning techniques.

We relied on MARBERT (Abdul-Mageed et al., 2021). Our choice was based on MARBERT robustness against dialectal Arabic as it was pre-trained on 15.6B tokens mainly tweets of multiple dialects. We use MARBERT as a sentence encoder to get an embedding that captures the semantic features of the text. We use the start token [CLS] vector as a representative of the entire sequence.

4.2 Contrastive Learning Phase

We apply a supervised contrastive learning approach similar to the on used for Natural Language Inference in (Gao et al., 2021).

First, we train our backbone on a contrastive objective. We adopted a supervised contrastive

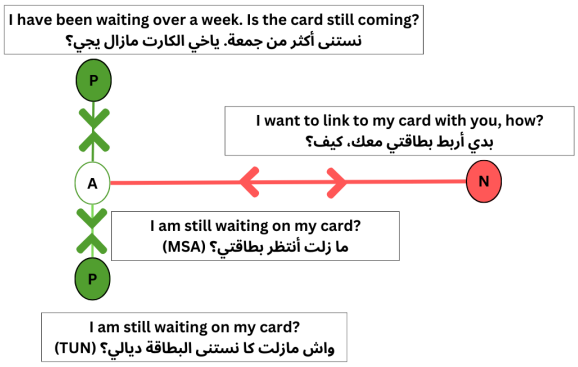

Figure 1: The triplet loss in action. The anchor and the positive samples are from *card_arrival* class and the negative sample from *card_linking* class.

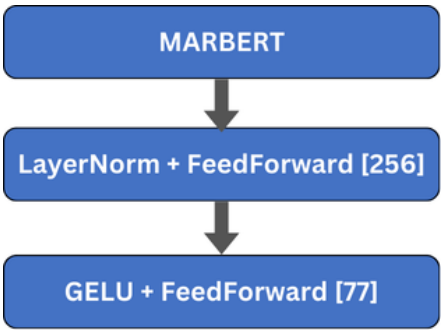

Figure 2: Our model architecture.

approach to identify positive and negative samples during the training process.

We created a sampling dictionary that maps each intent label to all queries of this intent in the training splits regardless of its dialect.

We used the triplet loss (Balntas et al., 2016) as our objective during training in this phase. This loss acts on three examples in each training step, namely, an anchor, a positive sample and a negative sample. The anchor is the current training sample. Positive sample is a random example from the anchor's class/intent. Negative sample is a random example from any class/intent other than the anchor's. The goal is to reduce the distance between the embedding of the positive sample and the anchor and increase it between the anchor and the negative sample in the feature space regardless of the dialect of each of them. Figure 1 demonstrates this procedure during training. Equation 1 shows how this loss is computed over a batch of N samples:

$$L_{triplet} = \frac{1}{N}\sum_{i=1}^{N} Max(0, d(a_i, p_i) - d(a_i, n_i) + \epsilon) \quad (1)$$

Where a_i, p_i and n_i are the anchor, positive, and negative samples respectively. $d(a, p)$ is the distance between a and p. ϵ is a non-negative margin representing the minimum difference between the positive and negative distances required for the loss to be 0.

4.3 Classification Model

After training the MARBERT encoder on the contrastive objective, we added feed-forward layers to map the output sentence embedding to one of the available intents. We used 2 linear layers with output dimensionality 256 and 77 and applied layer normalization (Ba et al., 2016) with GELU activation (Hendrycks and Gimpel, 2016) as nonlinearity. Figure 2 illustrates our architecture. This phase is trained using Cross-Entropy loss, equation 2, with class weights to ensure that the model will not be biased to dominant intents.

$$L_{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} w_c log\frac{exp(x_{n,c})}{\sum_{k=1}^{C} exp(x_{n,i})} y_{n,c} \quad (2)$$

4.4 Training Details

In the contrastive training phase, we trained the encoder for 200 epochs with a learning rate of 1e-5 and a 32 batch size. We tested the L2 distance and the cosine similarity as the distance function of the triplet loss. We set the margin ϵ to 1. After the training of this phase completes, we fine-tuned the model after adding the classification head for another 20 epochs using a batch size of 64 and a learning rate of 4e-6. We used reduce on plateau scheduler in both phases with a patience of 5 and a reduction factor of 0.5. All our training is done on a single NVIDIA V100 GPU.

5 Experiments

We report our experiments with the macro-F1 score, that treats every class equally, over all dialects available in the validation split. We will also report the micro-F1 score on the shared task's test split.

5.1 Number of Trainable Encoder Layers

We tested different number of trainable layers of MARBERTv1 model. We tested training only the classification layer [CLS], the last two encoder layers [2Ls], four layers [4Ls], and fine-tuning all the layers [Tot]. We applied this test with and without the contrastive pre-training phase and we used

Layers	w/o Contrastive	w/ Contrastive
CLS	0.8272	0.9090
2Ls	0.9088	0.9268
4Ls	0.9216	0.9263
Tot	0.9263	**0.9308**

Table 2: Macro-F1 score on the original validation split (MSA + PAL) when training different number of encoder layers with and without the contrastive pre-training phase.

Aug.	MSA	PAL	MOR	SAU	TUN
w/o	0.94	0.93	0.80	0.91	0.66
w/	**0.95**	**0.95**	**0.95**	**0.94**	**0.90**

Table 3: Per-dialect macro-F1 score before and after using the augmented part of the data.

the original data provided by the shared task. Table 2 shows the results on the validation split that contains MSA and PAL dialects only.

5.2 Effect of Data Augmentation

To assess the effect of the data augmentation, we retrained the entire encoder model using both contrastive and classification phases on the augmented dataset. This experiment is evaluated on the augmented validation split as well. The overall score across all dialects has improved from 0.86 to 0.94. We can see that the model had the worst performance on MOR and TUN dialects due to the huge dialectal differences they had with MSA compared to SAU that the model already had a good performance on. Detailed results are shown in table 3.

5.3 Distance Function Selection

The prior experiments used L2 distance in the contrastive pre-training phase. We tested the cosine similarity as well. We found that the new model performed better on some of the 77 intent classes. Hence we made a weighted average ensemble model using these two variants. Table 4 shows the difference between the two distance functions and the ensemble model.

5.4 Effect of Applying Pre-Processing

We tested whether applying some data cleaning techniques would improve our model performance. We mainly applied normalization and punctuation and diacritization removal. We retrained the same models again on the cleaned version of the data and compared the macro-F1 on the validation splits. The results are in Table 4.

Backbone	Cleaning Pipeline	L2 Dist.	Cos. Dist.	Ensemble
MARBERTv1	w/o	0.9377	0.9331	**0.9429**
	w/	0.9370	0.9355	**0.9428**
MARBERTv2	w/o	0.9357	0.9347	**0.9413**
	w/	0.9349	0.9368	**0.9417**

Table 4: Macro-F1 scores on our validation split of different variants of our method when using MARBERT v1/v2 and when analyzing the effect (with/without) data cleaning using two different distance functions and when combining the two distance functions models in an ensemble model.

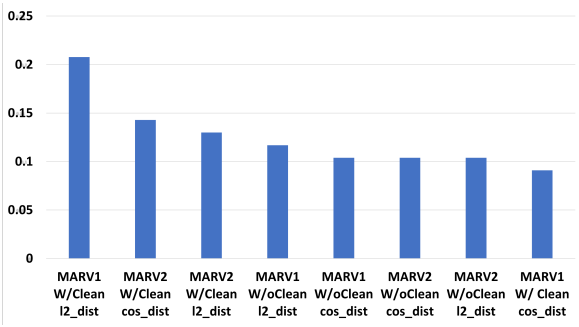

Figure 3: Percentage of intent classes where each model variant performs best.

5.5 Effect of Using MARBERTv2

Applying our cleaning pipeline didn't improve our performance. We tried using MARBERTv2 as our backbone instead of the first verison to enhance our model capabilities. This version was pre-trained on a larger corpus of data and longer sequence length. We trained the two versions using both the cleaned data and the original one and compared between the two versions in Table 4. This experiment has shown that using MARBERTv2 didn't improve the performance as expected.

According to table 4, we have eight variants of our model. We evaluated their performance by computing the per-class F1 score of each model and registered the best performing model for each intent. Figure 3 shows the percentage of intent classes which each model variant is best performing on. This indicates that an ensemble of all these model would have a better performance than using each model separately. We applied this ensemble and got 0.9487 on the validation split, outperforming our best result so far.

5.6 Qualitative Analysis

Table 5 shows some examples of misclassification in the validation splits. In the first example the

Example	Dialect	Input Query	Input Dialect	GT Intent	Prediction
Ex.1	MSA	أريد استرداد المبلغ على الخصم المباشر	SAU	direct debit payment not recognised	request refund
	SAU	ابغى استرجاع المبلغ على حسابي			
	ENG	I want a refund on a direct debit.			
Ex.2	MSA	لماذا تم تحصيل رسوم على التحويل الخاص بي؟	TUN	transfer fee charged	card payment fee charged
	SAU	علاش صاروا يطلبوا منى تكلفة؟			
	ENG	Why was my transfer charged a fee?			
Ex.3	MSA	كيف أتجنب الرسوم في المستقبل	TUN	card payment fee charged	direct debit payment not recognised
	TUN	كيفاش نجم نتجنب اتهامات في المستقبل؟			
	ENG	how do i avoid charges in the future			

Table 5: Misclassified samples from our validation split.

translated query didn't specify the card type being debit as in the original query, hence the model is correct for the given meaning that was changed from the ground truth. Also, in this example the ground truth didn't specify the refund intent of the user, which can be a take on the data annotators. In the second example the Tunisian translation didn't specify the type of fees charged, which are transfer fees. Lastly, the final example's translated query has a completely different meaning from the original one, which leads to misclassifying the user's intent.

5.7 Test Split Results

In this section, we are comparing different variants of our model that are pre-trained contrastively on the shared task test split. Table 6 shows the micro-F1 score of the different models we developed. The first row represent our first approach before adding the translated data. Data augmentation improved the model performance by 4.46% using the same modelling configuration. Changing the distance function to cosine distance decreased the micro-F1 by 0.6%. Then, we submitted using each ensemble model stated in table 4 and they all achieved close results. Finally, we used all versions in Table 4 in a single ensemble model and got our best result, 0.8763 micro-F1.

Model	Aug.	Clean.	Version	micro-F1
L2.dist	x	x	v1	0.8036
L2.dist	√	x	v1	0.8482
Cos.dist	√	x	v1	0.8422
Ens.1	√	x	v1	0.8586
Ens.2	√	√	v1	0.8588
Ens.3	√	√	v2	0.8696
Ens.4	√	√	v2	0.8721
Ens.All	√	√	both	**0.8763**

Table 6: Our results on the shared task's test split.

6 Discussion

Arabic is a morphologically rich language that has a large variation of dialects. This work shows the effectiveness of MARBERT pre-trained model as an Arabic dialect-agnostic model for intent classification. The implementation of contrastive learning further enhanced the model's ability to learn useful representations. Ensemble modelling further improved the performance. We can suggest using knowledge distillation to reduce model size while maintaining the high performance of the ensemble. The main limitation of this work is the dialect mismatch between the training and testing data and the absence of human annotated data. Hence, our model performance would be bound to the quality of the machine translation model we used to augment the data as seen in section 5.6.

7 Conclusion

This work investigated our work in subtask-1 of the AraFinNLP shared task 2024 by applying of contrastive learning for intent classification using two distance metrics. We used MARBERT as a backbone encoder model for Arabic dialects. Then, we added a classification module for intent prediction. Additionally, we explored the impact of enriching the data by translating MSA training data, applying data cleaning pipeline and using different versions of the backbone model. Lastly, outcomes of different model variants were combined in an ensemble model that achieved micro-f1 score on the competition's test set of 0.8762.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.

Waheed Ahmed Abro, Guilin Qi, Muhammad Aamir, and Zafar Ali. 2022. Joint intent detection and slot filling using weighted finite state transducer and bert. *Applied Intelligence*, 52(15):17356–17370.

Moath Al Ali, Bassel Zaity, Pavel Drobintsev, Hazem Wannous, Igor Chernoruckiy, and Andrey Filchenkov. 2020. Joint slot filling and intent detection in spoken language understanding by hybrid cnn-lstm model. In *Proceedings of the 2020 1st International Conference on Control, Robotics and Intelligent System*, pages 112–117.

Hala J. Alshahrani, Khaled Tarmissi, Hussain Alshahrani, Mohamed Ahmed Elfaki, Ayman Yafoz, Raed Alsini, Omar Alghushairy, and Manar Ahmed Hamza. 2022. Computational linguistics with deep-learning-based intent detection for natural language understanding. *Applied Sciences*, 12(17).

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. 2016. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Bmvc*, volume 1, page 3.

Naaima Boudad, Rdouan Faizi, Rachid Oulad Haj Thami, and Raddouane Chiheb. 2018. Sentiment analysis in arabic: A review of the literature. *Ain Shams Engineering Journal*, 9(4):2479–2490.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

Qinkai Chen. 2021. Stock movement prediction with financial news using contextualized embedding from bert. *arXiv preprint arXiv:2107.08721*.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

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, Minnesota. Association for Computational Linguistics.

Rehab Duwairi and Ftoon Abushaqra. 2021. Syntactic- and morphology-based text augmentation framework for arabic sentiment analysis. *PeerJ Computer Science*, 7:e469.

Rashmi Gangadharaiah and Balakrishnan Narayanaswamy. 2019. Joint multiple intent detection and slot labeling for goal-oriented dialog. 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 564–569, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Imane Guellil, Houda Saâdane, Faical Azouaou, Billel Gueni, and Damien Nouvel. 2021. Arabic natural language processing: An overview. *Journal of King Saud University - Computer and Information Sciences*, 33(5):497–507.

Changai He, Sibao Chen, Shilei Huang, Jian Zhang, and Xiao Song. 2019. Using convolutional neural network with bert for intent determination. In *2019 International Conference on Asian Language Processing (IALP)*, pages 65–70. IEEE.

Abdelrahman H. Hefny, Georgios A. Dafoulas, and Manal A. Ismail. 2020. Intent classification for a management conversational assistant. In *2020 15th International Conference on Computer Engineering and Systems (ICCES)*, pages 1–6.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv: Learning*.

Mustafa Jarrar, Ahmet Birim, Mohammed Khalilia, Mustafa Erden, and Sana Ghanem. 2023. Arbanking77: Intent detection neural model and a new dataset in modern and dialectical arabic. In *Proceedings of ArabicNLP 2023, Singapore (Hybrid), December 7, 2023*, pages 276–287. Association for Computational Linguistics.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *ArXiv*, abs/1609.01454.

Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. 2021. Finbert: A pre-trained financial language representation model for financial text mining. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence*, pages 4513–4519.

Sanad Malaysha, Mo El-Haj, Saad Ezzini, Mohammad Khalilia, Mustafa Jarrar, Sultan Nasser, and Ismail Berrada. 2024. AraFinNlp 2024: The first arabic financial nlp shared task. In *Proceedings of the 2nd Arabic Natural Language Processing Conference (Arabic-NLP), Part of the ACL 2024*. Association for Computational Linguistics.

Suman Ravuri and Andreas Stolcke. 2016. A comparative study of recurrent neural network models for lexical domain classification. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6075–6079. IEEE.

Ahmad Shapiro, Ayman Khalafallah, and Marwan Torki. 2022. AlexU-AIC at Arabic hate speech 2022: Contrast to classify. In *Proceedinsg of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection*, pages 200–208, Marseille, France. European Language Resources Association.

Rawan Tahssin, Youssef Kishk, and Marwan Torki. 2020. Identifying nuanced dialect for Arabic tweets with deep learning and reverse translation corpus extension system. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 288–294, Barcelona, Spain (Online). Association for Computational Linguistics.

Antonio Uva, Pierluigi Roberti, and Alessandro Moschitti. 2020. Dialog-based help desk through automated question answering and intent detection. *Computational Linguistics CLiC-it 2020*, page 443.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip Yu. 2019. Joint slot filling and intent detection via capsule neural networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5259–5267, Florence, Italy. Association for Computational Linguistics.