

# ArBanking77: Intent Detection Neural Model and a New Dataset in Modern and Dialectal Arabic

**Mustafa Jarrar**  
Birzeit University  
Birzeit, Palestine  
mjarrar@birzeit.edu

**Ahmet Birim**  
Sestek  
Istanbul, Türkiye  
ahmet.birim@sestek.com

**Mohammed Khalilia**  
Birzeit University  
Birzeit, Palestine  
mkhalilia@birzeit.edu

**Mustafa Erden**  
Sestek  
Istanbul, Türkiye  
mustafa.erden@sestek.com

**Sana Ghanem**  
Birzeit University  
Birzeit, Palestine  
swghanem@birzeit.edu

## Abstract

This paper presents the ArBanking77, a large Arabic dataset for intent detection in the banking domain. Our dataset was arabized and localized from the original English Banking77 dataset, which consists of 13,083 queries to ArBanking77 dataset with 31,404 queries in both Modern Standard Arabic (MSA) and Palestinian dialect, with each query classified into one of the 77 classes (intents). Furthermore, we present a neural model, based on AraBERT, fine-tuned on ArBanking77, which achieved an F1-score of 0.9209 and 0.8995 on MSA and Palestinian dialect, respectively. We performed extensive experimentation in which we simulated low-resource settings, where the model is trained on a subset of the data and augmented with noisy queries to simulate colloquial terms, mistakes and misspellings found in real NLP systems, especially live chat queries. The data and the models are publicly available at <https://sina.birzeit.edu/arbking77>.

## 1 Introduction

Intent detection falls under natural language understanding (NLU) and it aims at parsing the semantics of the user input in order to generate the best response. Intent representation is a mapping between the user request and the actions the chatbot triggers (Adamopoulou and Moussiades, 2020). Intent detection is typically considered a classification task, where each utterance is associated with one, and sometimes multiple, intents (Figure 1).



Figure 1: Examples queries and their intent.

Intent detection can be a challenging problem. The utterances during the chat are usually short,

providing only a brief context to rely on when predicting the intent and the label space can be very large requiring massive data annotation. In this paper, we present an Arabic intent dataset and a Bidirectional Encoder Representations from Transformers (BERT) based intent detection model.

The Arabic corpus presented in this paper is based on the Banking77, an English question-intent corpus for banking (Casanueva et al., 2020). Banking77 includes 13,083 queries, each query classified into one of the 77 intents. We first arabized the English Banking77 by providing an MSA version to each of the 13,083 queries, resulting in 15,537 MSA queries (some queries have more than one MSA variation). The arabization was done semi-automatically, first we used Google Translate and then manually verified and revised each query. Second, each query was manually re-written in the Palestinian dialect, resulting in 15,867 queries, which makes the data linguistically more representative from various aspects including phonology, morphology, lexicon, and syntax (Haff et al., 2022; Jarrar et al., 2017). The final dataset contains 31,404 queries, which was used to train a BERT-based model on intent detection task.

The rest of the paper is organized as follows: section 2 reviews the related work, section 3 presents the ArBanking77 corpus including data arabization and localization, section 4 presents the model architecture and training, section 5 presents the results for intent detection, section 6 presents our conclusion and section 7 states limitations.

## 2 Related Work

Arabic has a limited number of available labeled datasets, especially for dialectal and domain-specific tasks (Darwish et al., 2021; Naser-Karajah

et al., 2021). Due to data scarcity in Arabic language, research on Arabic intent detection is almost non-existent. Others have also stated the same, where conversational machine learning systems in Arabic are limited due to deficiency of datasets (Fuad and Al-Yahya, 2022) and Arabic conversational systems are lagging behind in applying the latest technology (Ahmed et al., 2022).

One of the closest work to Arabic intent detection is purposed in (Mezzi et al., 2022). The authors proposed intent detection model for the mental health domain in Tunisian Arabic. The idea is to classify the patient utterance or concern into five aspects: depression, suicide, panic disorder, social phobia and adjustment disorder. The data set was collected by simulating a real-life psychiatric interview where a 3D human avatar plays the doctor and asks the patient questions in Tunisian Arabic. The patient, in return, interacts with the avatar by answering the questions vocally, then the audio is transcribed to text. The authors used BERT as the encoder and added five binary classifiers, one classifier for each intent, achieving 0.94 F1 score.

Hijjawi et al. 2013 classified question and non-question utterances in chatbots. Decision trees were used to perform the classification and the model was integrated into ArabChat (Hijjawi et al., 2014) to classify utterances before processing them. Joukhadar et al. (2019) published a corpus in the Levantine Arabic dialect consisting of 873 sentences manually tagged with one of eight acts (greetings, goodbye, thanks, confirm, negate, ask/repeat, ask for alternative, and apology). The authors tried two features including Term Frequency-Inverse Document Frequency (TF-IDF) and  $n$ -gram. They also experimented with multiple classifiers and they concluded that Support Vector Machine (SVM) with 2-gram features performed the best at 0.86 accuracy.

Elmadany et al. (2018) introduced a speech-act recognition and sentiment dataset (ArSAS). About 21K tweets were collected and manually labeled with two types of classes: speech-act and sentiment. Speech-act labels include expression, assertion and question, while the sentiment labels are negative, positive, neutral and mixed. Algotiml et al. (2019) trained two models on the ArSAS dataset, a Bi-directional Long-Short Term Memory (BiLSTM) and SVM and achieved an accuracy of 0.875 and a macro F1 score of 0.615. (Zhou et al., 2022) proposed a contrastive based learning for out-of-

domain data and tested the performance on multiple datasets including the Banking data (Casanueva et al., 2020) and they demonstrated improvement of the out-of-domain data without sacrificing performance on in-domain-data.

Other related languages for which intent detection was studied is Urdu. In (Shams et al., 2019), the authors translated the Air Travel Information System (ATIS) (Hemphill et al., 1990) and AOL datasets from English to Urdu and performed intent detection using a combination of CNNs, LSTMs and BiLSTMs models. For ATIS, CNN performed the best at 0.924 accuracy, while for AOL, BiLSTM achieved the highest performance at 0.831 accuracy. In later work the authors improved the accuracy to reach 0.9112 (Shams and Aslam, 2022). ATIS was also used for intent detection in the Indonesian language (Bilah et al., 2022) and the authors reported an accuracy of 0.9584 using a CNN-based model. (Basu et al., 2022) utilized Snips (Coucke et al., 2018) and ATIS to train a meta-learning approach with contrastive learning for intent detection and slot-filling. Snips dataset covers multiple domains including restaurants, books, weather and music, making it more challenging than ATIS. The data is collected using Snips personal assistant and contains 16K queries labeled with 7 intents.

The reader may have already noticed that we could not find relevant work related to Arabic intent detection recognition or any related work on labeled Arabic intent datasets. In this paper, we attempt to address these two issues, Arabic intent corpus and intent recognition. We present the ArBanking77, an Arabic intent dataset, which was arabized and localized from the Banking77 English dataset (Casanueva et al., 2020). ArBanking77 was also augmented with thousands of additional MSA and Palestinian dialect queries, resulting in a final dataset of 31,404 queries and 77 intents. ArBanking77 was used to fine-tune BERT-based model, achieving an F1-score of 0.9209 and 0.8995 on MSA and Palestinian dialect, respectively.

When deploying a fine-tuned intent detection model inside a chatbot system, other modules might be needed to better understand user queries, such as spell corrections (Eryani et al., 2020), named entity recognition (Jarrar et al., 2022; Liqreina et al., 2023), word-sense disambiguation (Al-Hajj and Jarrar, 2021; Jarrar et al., 2023a), synonymy expanding (Ghanem et al., 2023; Jarrar et al., 2021).

### 3 The ArBanking77 Corpus

The ArBanking77 corpus is derived from the Banking77 dataset (Casanueva et al., 2020) that consists of 13,083 queries and 77 classes (intents) and that is open under the (CC-BY-4.0) license. Banking77 was designed to focus on a fine-grained single domain, banking. Each query is labeled with one of the 77 classes. Example intents from the dataset include *card arrival*, *Personal Identification Number (PIN) blocked*, *card linking*, *exchange rate* and *age limit*. The number of queries per class ranges between 75 to 227, with an average of 170 queries per intent. The original Banking77 dataset is divided into train and test dataset, their statistics are presented in Table 1.

	Train Set	Test Set
Query count	10,003	3,080
Avg word count	11.95	10.95
Min word count	2	2
Max word count	79	69
Std of word count	7.89	6.69

Table 1: Statistics of the Banking77 English dataset

Banking77 was arabized and localized into ArBanking77 by 26 annotators through multiple phases and over several months. Each query in the Banking77 has at least two corresponding queries in the ArBanking77 (at least one query written in each MSA and Palestinian dialect).

#### 3.1 Phase I: Arabization and Localization

The first step was the translation of the Banking77 from English into MSA. We used Google Translate API to translate the 13,083 queries. For each original English query,  $j$ , where  $0 < j < m$  and  $m = 13,083$ , we form the following tuple:

$$(q_j^i, q_j^{En}, q_j^{MSA_1}, q_j^{MSA_2}, q_j^{PAL_1}, q_j^{PAL_2})$$

$$\forall 0 < j < m$$

where  $q_j^i$  is the query’s intent,  $q_j^{En}$  is the original English query from Banking77,  $q_j^{MSA_1}$  is the MSA translation,  $q_j^{MSA_2}$  is a second MSA query,  $q_j^{PAL_1}$  is the Palestinian query, and  $q_j^{PAL_2}$  is a second Palestinian query.

Each annotator was asked to understand the English query and its intent, then: (i) review  $q_j^{MSA_1}$ , and revise it if needed; (ii) optionally write  $q_j^{MSA_2}$ , (iii) write a  $q_j^{PAL_1}$  query, and (iv) optionally write a  $q_j^{PAL_2}$  query. The annotators performed these

steps according to the following arabization and localization guidelines:

- $q_j^{MSA_1}$  should be revised in case of incorrect translation. We also ensured the translation is adapted to the banking domain. For example, *transfer* was incorrectly translated into *نقل* *lnaql* (ship) instead of *تحويل* *lthwyl* (money transfer); *activate* was translated to *تنشيط* *tnšyṭ*, which is not semantically wrong, but it should be *تفعيل* *tafʿyl*, as it is the common term used in the banking domain. The total number of revised translations is 2,104 (~ 16%).
- $q_j^{MSA_2}$  is optionally written by the annotator if there is a need to add an extra formulation of the MSA query. For example, Personal Identification Number might be translated in  $q_j^{MSA_1}$  as *(رقم التعريف الشخصي)* and *(الرقم السري)* as a second formulation in  $q_j^{MSA_2}$ .
- $q_j^{PAL_1}$  is the formulation of the query in the Palestinian dialect, reflecting the terminology Palestinians naturally use in banking services.
- $q_j^{PAL_2}$  is optionally written by the annotator if there is a need to add an extra formulation of the query in the Palestinian dialect.

This phase was carried out by 26 annotators, who are 3<sup>rd</sup> and 4<sup>th</sup> year college students. Each annotator was given about 500  $q_j^{En}$  queries and their translations ( $q_j^{MSA_1}$ ) to revise. Based on  $q_j^{En}$  and  $q_j^{MSA_1}$ , annotators also provided  $q_j^{MSA_2}$ ,  $q_j^{PAL_1}$ , and  $q_j^{PAL_2}$ . When generating PAL queries, annotators had access to both English and MSA queries, which may bias the PAL query towards MSA. However, we verified that this is not a concern as the lexical overlap between MSA and PAL is significant (Section 3.3). Furthermore, in order to diversify the queries, we avoided having all queries in one intent reviewed and written by one annotator only. Instead, each intent was divided among multiple annotators, usually 2-5 annotators.

#### 3.2 Phase II: Review

To control and verify the quality of the data generated in Phase I, we performed a final manual review. Each of the 26 annotators, employed for phase I, was assigned a set of queries to review. On average three intents were assigned to each reviewer

and we ensured that all queries belonging to one intent are assigned to the same reviewer. In order to increase data labeling consistency, we added the constraint that classes assigned to one reviewer should be relevant to each other (i.e., card arrival, card linking, card activation). Each reviewer was asked to pay attention to the following issues: (i) The MSA and Palestinian queries should be acceptable, semantically correct and well-formulated; (ii) all queries in one intent belong to that intent, and not to other intents (labeling consistency); and (iii) spelling mistakes are ignored in order to simulate common errors and noise in real NLP systems, especially in live chat queries.

Once the review is complete, we revised duplicate queries by introducing additional variations to make them unique. Duplicate queries can arise when we have many-to-one translations, in other words, multiple English queries are translated into one Arabic query (see examples in Table 2).

Our final ArBanking77 dataset (Table 3) consists of 31,404 queries in total, 2.4x larger than the Banking77 dataset. On average, there are 408 queries per intent (202 MSA queries/intent and 206 Palestinian queries/intent). We further divided our training data into train and validation sets, by sampling 90% of the queries in the  $i$ th class to the training set and the remaining 10% were included in the validation set. This is contrary to the train/test only split cited in (Casanueva et al., 2020), in which they stated small data size as the reason for not introducing a validation set.

Table 4 presents some statistics about ArBanking77. From Table 4 we observe that the dialectal queries are shorter than their corresponding MSA queries. In MSA the average number of words in a query is 9.85, while it is 8.06 in the Palestinian queries. This is expected as in some cases dialectical Arabic omits interrogative nouns such as (هل), so an MSA query such as (هل يوجد شروط للعمرة?) is phrased in Palestinian dialect as (في شروط علمعمة?). In other cases, functional words such as prepositions (عن/from or about, على/on or above, إلى/to or at, في/in or into) are used as prefixes or suffixes. For instance, the phrase (على العمرة) in MSA is (علمعمة /*ʿumr*) in the Palestinian dialect, where (على) is used as a prefix in the word (علمعمة /*ʿumr*). For discussion on the orthography of Arabic dialect, see (Nayouf et al., 2023; Haff et al., 2022; Jarrar et al., 2014)

### 3.3 Lexical Relation between MSA and PAL

Arabic is a highly diglossic language, meaning that two or more distinct languages are spoken within a given region, which is a phenomenon in the Arab countries (Jarrar, 2021). Sometimes MSA is significantly different from colloquial dialects (Jarrar et al., 2023b; Naser-Karajah et al., 2021), where they can be mutually unintelligible. Because of that MSA and PAL have many differences making it harder to apply MSA NLP tools to PAL. In this section, we will study the lexical difference between MSA and PAL, although the differences extend beyond lexical to include morphology, phonology, orthography, semantic and syntactic.

To measure the lexical overlap between MSA and PAL, we computed the Jaccard Index for each parallel pair (MSA and PAL) and averaged the results across the entire dataset. We found that the mean Jaccard index is 0.16, median 0.13 and standard deviation 0.13. Others have also studied the lexical overlap between MSA and PAL and reported similar results. For instance, (Kwaik et al., 2018) measured the overlap between MSA and other dialects including PAL on two parallel datasets, the Parallel Arabic Dialect Corpus and Multi-Dialectal Arabic and reported Jaccard Index of 0.19 and 0.16, respectively. This shows that for diglossic languages such as Arabic, training on one variation is not necessarily extensible. Later in section 5.1, we will explore zero-shot learning to illustrate the effect of lexical differences on model performance.

## 4 Intent Detection Model

We fine-tuned a BERT-based model on an intent detection task using the ArBanking77 dataset. In this section, we will go over the model details.

### 4.1 Model Architecture

Our model is based on BERT, a transformer-based language representation for natural language processing (Devlin et al., 2018). BERT was developed by Google in 2018 as a solution for the most common language tasks such as sentiment analysis, named entity recognition, and question answering. BERT is built using transformers, which is a deep learning architecture that solves sequence-to-sequence tasks in NLP and relies on the attention mechanism that learns the alignment between words in a given sequence. Transformers include two components: an encoder that encodes the input



English Queries	Arabic Query
Can you tell me the restrictions for the disposable cards?	هل يمكنك إخباري بالتبويد المفروضة على البطاقات التي تستخدم مرة واحدة.
Can you please inform me of the restrictions for the disposable cards.	
How is an exchange rate calculated?	كيف يتم حساب سعر الصرف؟
How are your exchange rates calculated?	

Table 2: Examples of many-to-one English-Arabic translation.

	MSA ( $q_n^{MSA_1} + q_n^{MSA_2}$ )	PAL ( $q_n^{PAL_1} + q_n^{PAL_2}$ )	Total
Train	10,733	10,826	21,559
Validation	1,230	1,234	2,464
Test	3,574	3,807	7,381
Total	15,537	15,867	<b>31,404</b>

Table 3: Size of ArBanking77

	MSA	PAL	Overall
Avg word count	9.85	8.06	8.95
Std of word count	6.54	4.66	5.74
Min word count	2	2	2
Max word count	68	54	68

Table 4: Statistics of ArBanking77 dataset

text and a decoder that produces a prediction for the task, such as predicting masked token or predicting next sentence. In this paper, BERT encoder is fine-tuned on Arabic intent detection task using the ArBanking77 dataset.

For intent detection, a single linear layer was added on top of BERT transformer layers to perform the intent classification task.

## 4.2 Model Training

We fine-tuned multiple pre-trained transformer models, which will be discussed in the next section. The hyperparameters we used are: learning rate,  $1e^{-3} < \eta < 5e^{-5}$ , and batch size,  $B = \{16, 32, 64\}$ . We ran approximately 30 experiments, with an average run-time per experiment  $< 2$  hours, depending on model parallelism. The best performing hyperparameters were  $\eta = 4e^{-5}$  and  $B = 64$ , with maximum sequence length of 128, maximum of 20 epochs and early termination if there is no improvement on the validation data after five epochs. Model training was performed using our Nvidia Tesla P100 16GB GPU card.

## 5 Experiments and Results

We ran multiple experiments with different models and data configurations. In section 5.1, we evaluate zero-shot learning, section 5.2 benchmarks multiple pre-trained transformer models on Arabic data, section 5.3 simulates low-resource settings

and section 5.4 simulates different spelling errors that are commonly found in the Arabic language. We report the model performance on the test set using macro F1, precision and recall scores.

When training the models on the full dataset, we used the train, validation and test split listed in Table 3, where 21,559 queries used for training and 2,464 served as the validation set. In low-resource settings we experimented with different training and validation data sizes (Section 5.3), but the test set size remained at 7,381 queries. In noise and error simulation experiments we used the same test set with 7,381 queries, but errors were injected into the test queries as we will explain in Section 5.4.

### 5.1 Zero-Shot Cross-Lingual Transfer Learning

In some cases, zero-shot cross-lingual transfer learning can yield good results and may help us avoid the manual data annotations. In this section, we study how zero-shot cross-lingual transfer learning perform on both MSA and PAL using multi-lingual BERT (mBERT) (Devlin et al., 2018) and GigaBERT (Lan et al., 2020). mBERT is trained on 104 languages including Arabic, which is based on MSA data from Wikipedia with less than 1.4 gigabytes and only 7,292 tokens (Alammery, 2022). GigaBERT was trained for Arabic NLP tasks and English-to-Arabic zero-shot transfer learning. The data contained about 13 million articles from different sources and augmented with code-switched samples to improve cross-lingual learning.

In one set of experiments we evaluated zero-shot cross-lingual transfer learning on PAL test set by fine-tuning mBERT on ArBanking77 MSA training dataset, which yielded 0.5968 F1-score (Table 5). In the second set of experiments we

performed zero-shot cross-lingual transfer learning on both MSA and PAL by fine-tuning GigaBERT and mBERT on the English Banking77 training data. On MSA, GigaBERT and mBERT achieved 0.5047 and 0.1774 F1-score, respectively. The performance is even lower on PAL with GigaBERT and mBERT performing at 0.3507 and 0.0903 F1-score, respectively. These experiments demonstrate the performance of multilingual pre-trained models falls behind on MSA and is significantly lower for dialectical Arabic, which begs the need for MSA and dialectical Arabic data annotations.

## 5.2 Pre-Trained Transformers Benchmark

As we observed in the previous section, multilingual pre-trained transformers did not perform well on MSA and PAL. In this section, we evaluate various Arabic pre-trained transformer models in addition to mBERT on ArBanking77 dataset. We benchmark against the following models:

*AraBERT* (Antoun et al., 2020): trained on two major datasets, Abu El-Khair, a 1.5B words Arabic Corpus (El-Khair, 2016) and the Open Source International Arabic News Corpus (OSIAN), which consists of 3.5 million articles (1B tokens), from 31 news sources in 24 Arab countries (Zeroual et al., 2019). The final size of AraBERT dataset is 70M sentences, corresponding to about 24GB of text.

*ARBERT* (Abdul-Mageed et al., 2021): trained on 61GB (6.5B tokens) of MSA text in books, news articles, Gigaword (Parker et al., 2011), Open Super-large Crawled Almanach corpus (OSCAR) (Ortiz Suárez et al., 2019), OSIAN and the Wikipedia Arabic (Attardi, 2015).

*MARBERT* (Abdul-Mageed et al., 2021): trained on dialectical Arabic collected from Twitter.

*MARBERTv2* (Abdul-Mageed et al., 2021): trained on the ARBERT MSA data in addition to dialectical Arabic, has longer sequence length, trained for more epochs and contains a total of 29B tokens.

*QARiB* (Abdelali et al., 2021): Qatar Computing Research Institute (QCRI) Arabic and Dialectal BERT trained on Arabic Gigaword Fourth Edition (1B words), Abu El-Khair Corpus (1.5B words) and Open Subtitles (0.5B words).

*CAMeLBERT-Mix* (Inoue et al., 2021): trained on a mix of MSA data that includes Gigaword Fifth Edition, Abu El-Khair Corpus, OSIAN, Arabic Wikipedia, OSCAR, dialectical Arabic that covers Levantine and Gulf regions, and a subset of the OpenITI corpus (Nigst et al., 2020)

Results for those models are presented in Table 6, sorted by the PAL test F1-score. AraBERTv2 gives the best F1-score on both MSA and PAL with 0.9209 and 0.8995, respectively. In the remaining experiments, we will use AraBERTv2 given that it achieved the best performance.

Those results are based on fine-tuning the models on the manually reviewed translations. To see if the manual review of the translations improves the model performance we fine-tune two additional AraBERTv2 models. One using the original machine translated data and the second with the manually reviewed data. Note that both training datasets contain MSA only data, since Google Translate will produce MSA translation. Fine-tuning with the original translations results in F1-scores of 0.9099 and 0.7945 for MSA and PAL, respectively. When the data is manually reviewed the F1-scores are 0.9117 and 0.7918 for MSA and PAL, respectively. A very small difference, yet it was important to review the translations to adapt it to the banking domain.

## 5.3 Low-Resource Simulation

This section aims to investigate the impact of the size of the training set on the model performance. Since data labeling is typically expensive it is important to estimate the number of samples one needs to achieve good and acceptable accuracy. We conducted several experiments with different training data sizes: 20% (of the training queries per intent were randomly sampled), 50% and 100% (the entire training set). Throughout all the experiments, we evaluated our model on same test set, which contains 7,381 queries.

Results with different low-resource settings are presented in Table 7. The average increase in F1-score as we increase the training data size is about 2.26% and 3.16% on the MSA and PAL test datasets, respectively, which indicates the impact of the training dataset size is more noticeable on the dialectical Arabic. We also notice that the performance on the PAL test is consistently lower than MSA test. The performance gap between MSA and PAL is 2.14%, 2%, and 3.95% F1-score when training with 100%, 50% and 20% of the data, respectively. The largest performance gap between MSA and PAL is at the lowest setting (20%), after that the performance gap stabilizes. Lower performance on dialectical data could be due AraBERT (Antoun et al., 2020) not being sufficiently exposed to the

Pre-trained Model	Training Data	MSA F1	PAL F1
Multi-lingual BERT (uncased)	ArBanking77 (MSA)	-	0.5968
GigaBERT	Banking77 (English)	0.5047	0.3507
Multi-lingual BERT (uncased)	Banking77 (English)	0.1774	0.0903

Table 5: Performance of zero-shot learning.

Pre-trained Model	MSA Test			PAL Test		
	Precision	Recall	F1	Precision	Recall	F1
AraBERTv2	<b>0.9231</b>	<b>0.9212</b>	<b>0.9209</b>	<b>0.9004</b>	<b>0.9025</b>	<b>0.8995</b>
MARBERTv2	0.9161	0.9142	0.9138	0.8983	0.8981	0.8962
ARBERT	0.9103	0.9121	0.9115	0.8810	0.8923	0.8899
QARiB	0.9147	0.9123	0.9121	0.8846	0.8864	0.8835
CAMeLBERT-Mix	0.9149	0.9133	0.9128	0.8855	0.8854	0.8830
MARBERT	0.9106	0.9075	0.9070	0.8817	0.8817	0.8789
Multi-lingual BERT	0.8888	0.8872	0.8862	0.8598	0.8623	0.8578

Table 6: Performance of various pre-trained transformers on ArBanking77

Palestinian dialect during the pretraining phase. In general, dialectical Arabic is typically noisier and does not follow consistent orthography as MSA.

Surprisingly, the performance on the MSA and PAL test sets using only 20% of the training data is impressive at 0.8758 and 0.8363 F1-scores, respectively. This indicates that we can expect to achieve an acceptable performance on other low-resource dialectical Arabic on intent detection task.

#### 5.4 Noise and Error Simulation

Colloquial words, misspellings and different word variations present a challenge to chatbots. Therefore, in this section we aim to measure the robustness of our dataset and model. We experimented with three types of error and noise simulations: (1) common spelling errors ( $sim_c$ ), (2) simulated errors ( $sim_s$ ), and (3) keyboard-related errors ( $sim_k$ ) - see Appendix A for the details.

We performed experiments with and without training data augmentation. In case of augmentation, train and test sets were augmented in slightly different fashion. For training, about 50% of the queries were augmented with  $sim_s$  and the other 50% were augmented with  $sim_k$ . The original data was combined with the augmented data resulting in 43,118 queries in the training set. We evaluated the model on three versions of the test set, one version injected  $sim_c$  errors in each query, the second version using  $sim_s$  and the third with  $sim_k$ .

Results of the combined low-resource and error simulations are summarized in Table 8. Due to the number of experiments, we only reported the macro F1-score. We see a similar trend to the results presented in Section 5.3, the model performance on the PAL test set is consistently lower than MSA test

set across all experiments. We also notice that the model is more sensitive to some errors introduced into the test set.

We performed the experiments using two trained models, with and without training augmentation. In both models we see similar behaviour, where we observe that the average drop in performance, when reducing training set size, on PAL- $sim_c$  across all data settings is about 3.38%, compared to 2.37% on MSA- $sim_c$ . Similar pattern is also observed on the PAL- $sim_s$  and MSA- $sim_k$ , with an average performance drop of 3.39% and 2.16%, respectively. However, we see a lower performance on PAL- $sim_s$  with an average drop in F1-score by 4.2%, compared to 2.19% on MSA- $sim_s$ . From that, we learn that the model performance is stable on MSA regardless of the type of errors we inject into the data, however, on PAL we see more volatility and sensitivity in the model performance when injecting  $sim_s$  errors. Those findings reveal that BERT is more susceptible to the removal of spaces in dialectical Arabic since that results in combining two or three tokens into one. This issue is exacerbated further in dialectical Arabic since it lacks consistent orthography compared to MSA.

Despite those results, we see that augmenting the training data did help close the performance gap between the PAL and MSA. Figure 2 zooms in a little more into the performance on MSA- $sim_s$  and PAL- $sim_s$  with and without training augmentation. Three observations to make from Figure 2: 1) MSA performance is better than PAL regardless of data augmentation, 2) augmenting the training data closes the performance gap between PAL- $sim_s$  (augmented) and MSA- $sim_s$  (without augmentation), 3) the average F1-score gain after training

% of data	MSA Test			PAL Test		
	Precision	Recall	F1	Precision	Recall	F1
20%	0.8825	0.8755	0.8758	0.8441	0.8403	0.8363
50%	0.9117	0.9094	0.9088	0.8909	0.8903	0.8888
100%	0.9231	0.9212	0.9209	0.9004	0.9025	0.8995

Table 7: Results on the ArBanking77 MSA and PAL test sets in low-resource settings

with augmented data on PAL- $sim_s$  (4.12%) is larger than MSA- $sim_s$  (2.2%). The improvements are less noticeable on  $sim_c$  and  $sim_k$ .

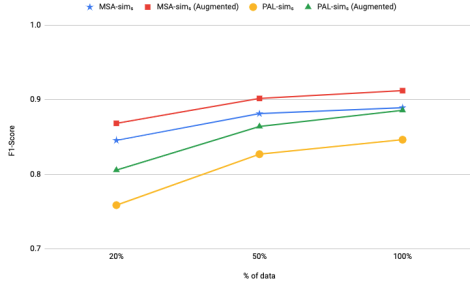


Figure 2: MSA- $sim_s$  vs. PAL- $sim_s$  F1-scores with low-resource settings, (Augmented) indicates that the training data was augmented.

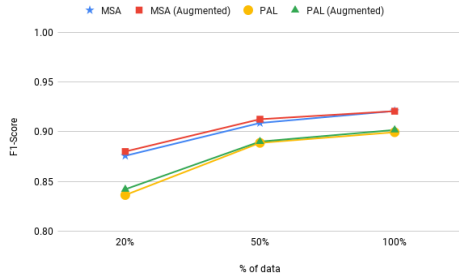


Figure 3: MSA vs. PAL clean sets F1-scores with low-resource settings and data augmentation, (Augmented) indicates that the training data was augmented.

Figure 3 shows that training data augmentation does not affect the performance on the clean MSA and PAL test sets. On the contrary, at the lowest resource settings the augmented model outperformed the non-augmented on MSA and PAL by 0.43% and 0.58%, respectively. At 50% and 100% settings, both the augmented and non-augmented models' performance converge on MSA and PAL.

## 6 Conclusion

In this paper, we presented the ArBanking77 dataset, consisting of queries in both MSA and Palestinian dialects in the banking domain. As far as we know, ArBanking77 is the first Arabic intent detection dataset in the banking domain. The

dataset contains 31,404 queries and 77 intents. The data was then used to fine-tune a BERT-based model for the intent detection task, resulting in an F1-score of 0.9209 for MSA and 0.8995 for PAL. We also simulated low-resource settings and found that the model is robust and with only 20% of the data, model performance on PAL and MSA dropped by only 6.32% and 4.51%, respectively. We noted that training data augmentation does not negatively affect the model performance on the clean MSA and PAL test sets. In fact, at the lowest resource settings (20%) the augmented model outperformed the non-augmented model on both MSA and PAL.

We performed additional data augmentation to simulate errors, misspellings, and other mistakes that are common in real NLP systems. We observed the accuracy on PAL- $sim_s$  suffers greatly when the model is trained on 20% of the non-augmented data. Augmenting the training data closes the performance gap on PAL- $sim_s$  by about 5%. This indicates that BERT is susceptible to some errors, especially in dialectal Arabic which has less consistent orthography than MSA. It is also noticeable that the relative drop in accuracy between the 20% and 50% training sets is much larger than 50% and 100% case. This implies that the negative effect of the introduced errors in the dialectal Arabic is inversely proportional to the amount of data used in the train set. Finally, based on the low performance using zero-shot learning on MSA and PAL and a slight lexical overlap between them, we concluded that there is an urgent need to annotate MSA and dialectal Arabic.

## 7 Limitations

Our dataset is limited to MSA and Palestinian dialect and covers only 77 intents. Applying our models and data to dialects others than MSA and PAL may not yield accurate intents. Furthermore, our data covers intents that are commonly found in traditional banking. Additional intents may need to be studied from non-traditional banking such as Islamic banks. We plan to extend our dataset to



Train Augmentation	Test Augmentation	MSA Test			PAL Test		
		20%	50%	100%	20%	50%	100%
None	None	0.8758	0.9088	0.9209	0.8363	0.8888	0.8995
	$sim_c$	0.8452	0.8795	0.8981	0.7933	0.8435	0.8637
	$sim_s$	0.8454	0.8813	0.8893	0.7585	0.8269	0.8463
	$sim_k$	0.8392	0.8648	0.8844	0.7942	0.8428	0.8634
$sim_s/sim_k$	None	0.8801	0.9126	0.9207	0.8421	0.8901	0.9018
	$sim_c$	0.8583	0.8922	0.9001	0.8065	0.8602	0.8711
	$sim_s$	0.8683	0.9017	0.9121	0.8055	0.8641	0.8857
	$sim_k$	0.8499	0.8833	0.8909	0.8086	0.8529	0.8749

Table 8: Performance in terms of F1-scores of models trained on the combined MSA and PAL datasets when simulating low-resource setting (20% of the data) and different types of noise, "None" refers to the clean dataset while the percentages in the header indicate the percentage of training data used.

cover more Arabic dialects and obtain data from non-traditional banking institutions in the Arab region to better understand the difference in intents compared to the traditional banking. Moreover, we want to explore natural language understanding in the banking domain by combining named entity recognition with intent detection.

We can further improve model performance by adding additional auxiliary loss functions such as contrastive loss, which will help align the token representations between the MSA and PAL queries. Furthermore, due to data limitation, the models trained on the data, including Banking77, perform intent classification using a single utterance. In practice, the query has a context, preceding utterances, that can provide important signal to the model, which may lead to better performance.

## Acknowledgements

This research is partially funded by the Palestinian Higher Council for Innovation and Excellence. We would like to thank Taymaa Hammouda for the technical support. The authors also acknowledge the great efforts of many students who helped in the annotation process, especially Rania Shawsan, Dalal Bawatneh, Malak Elsheik, Maissan Qadi, Marah Beirat, Sara Shaabna, Hiba Qasrawee, Manar Jawabreh, Mohammad AbuBader, Tuqa Qurt, Rahaf Bakeer, Marah Qoud, Nirmeen Al-Sheikh, Ameer Eleyan, Sondus Ilawi, Fatima Kusbeh, Ameena Jadallah, Eyab Ghifari, Lina Salameh, Shorouq Zaid, Mariam Abdelqader, Noor Momani, Yazan Assaf, Hala Hamza, Sondus Majdobe and Sawsan Mohamad.

## References

Ahmed Abdelali, Sabit Hassan, Hamdy Mubarak, Kareem Darwish, and Younes Samih. 2021. [Pre-training](#)

[bert on arabic tweets: Practical considerations.](#)

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.

Eleni Adamopoulou and Lefteris Moussiades. 2020. An overview of chatbot technology. In *Artificial Intelligence Applications and Innovations*, pages 373–383, Cham. Springer International Publishing.

Arfan Ahmed, Nashva Ali, Mahmood Alzubaidi, Wajdi Zaghoulani, Alaa Abd-alrazaq, and Mowafa Househ. 2022. [Arabic chatbot technologies: A scoping review.](#) *Computer Methods and Programs in Biomedicine Update*, 2:100057.

Moustafa Al-Hajj and Mustafa Jarrar. 2021. [Arabglossbert: Fine-tuning bert on context-gloss pairs for wsd.](#) In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 40–48, Online. INCOMA Ltd.

Ali Saleh Alammary. 2022. [Bert models for arabic text classification: A systematic review.](#) *Applied Sciences*, 12(11).

Bushra Algotiml, AbdelRahim Elmadany, and Walid Magdy. 2019. [Arabic tweet-act: Speech act recognition for Arabic asynchronous conversations.](#) In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 183–191, Florence, Italy. Association for Computational Linguistics.

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, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Giuseppe Attardi. 2015. Wikiextractor. <https://github.com/attardi/wikiextractor>.

- Samyadeep Basu, Amr Sharaf, Karine Ip Kiun Chong, Alex Fischer, Vishal Rohra, Michael Amoake, Hazem El-Hammamy, Ehi Nosakhare, Vijay Ramani, and Benjamin Han. 2022. [Strategies to improve few-shot learning for intent classification and slot-filling](#), pages 17–25. Association for Computational Linguistics.
- Chiva Olivia Bilah, Teguh Bharata Adji, and Noor Akhmad Setiawan. 2022. [Intent detection on indonesian text using convolutional neural network](#). In *2022 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*, pages 174–178.
- 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.
- Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.
- Kareem Darwish, Nizar Habash, Mourad Abbas, Hend Al-Khalifa, Husein T. Al-Natsheh, Houda Bouamor, Karim Bouzoubaa, Violetta Cavalli-Sforza, Samhaa R. El-Beltagy, Wassim El-Hajj, Mustafa Jarrar, and Hamdy Mubarak. 2021. [A panoramic survey of natural language processing in the arab worlds](#). *Commun. ACM*, 64(4):72–81.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Ibrahim Abu El-Khair. 2016. 1.5 billion words arabic corpus. *arXiv preprint arXiv:1611.04033*.
- AbdelRahim Elmadany, Hamdy Mubarak, and Walid Magdy. 2018. Arsas: An arabic speech-act and sentiment corpus of tweets. *OSACT*, 3:20.
- Fadhl Eryani, Nizar Habash, Houda Bouamor, and Salam Khalifa. 2020. A spelling correction corpus for multiple arabic dialects. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4130–4138.
- Ahlam Fuad and Maha Al-Yahya. 2022. [Recent developments in arabic conversational ai: A literature review](#). *IEEE Access*, 10:23842–23859.
- Sana Ghanem, Mustafa Jarrar, Radi Jarrar, and Ibrahim Bounhas. 2023. [A benchmark and scoring algorithm for enriching arabic synonyms](#). In *Proceedings of the 12th International Global Wordnet Conference (GWC2023)*, pages 215–222. Global Wordnet Association.
- Karim El Haff, Mustafa Jarrar, Tymaa Hammouda, and Fadi Zaraket. 2022. [Curras + baladi: Towards a levantine corpus](#). In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2022)*, Marseille, France.
- Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. [The ATIS spoken language systems pilot corpus](#). In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.
- Mohammad Hijjawi, Zuhair Bandar, and Keeley Crockett. 2013. [User’s utterance classification using machine learning for arabic conversational agents](#). In *2013 5th International Conference on Computer Science and Information Technology*, pages 223–232.
- Mohammad Hijjawi, Zuhair Bandar, Keeley Crockett, and David Mclean. 2014. [Arabchat: An arabic conversational agent](#). In *2014 6th International Conference on Computer Science and Information Technology (CSIT)*, pages 227–237.
- Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, Kyiv, Ukraine (Online). Association for Computational Linguistics.
- Mustafa Jarrar. 2021. [The arabic ontology - an arabic wordnet with ontologically clean content](#). *Applied Ontology Journal*, 16(1):1–26.
- Mustafa Jarrar, Nizar Habash, Diyam Akra, and Nasser Zalmout. 2014. [Building a corpus for palestinian arabic: a preliminary study](#). In *Proceedings of the EMNLP 2014, Workshop on Arabic Natural Language*, pages 18–27. Association For Computational Linguistics.
- Mustafa Jarrar, Nizar Habash, Faeq Alrimawi, Diyam Akra, and Nasser Zalmout. 2017. [Curras: An annotated corpus for the palestinian arabic dialect](#). *Journal Language Resources and Evaluation*, 51(3):745–775.
- Mustafa Jarrar, Eman Karajah, Muhammad Khalifa, and Khaled Shaalan. 2021. [Extracting synonyms from bilingual dictionaries](#). In *Proceedings of the 11th International Global Wordnet Conference (GWC2021)*, pages 215–222. Global Wordnet Association.
- Mustafa Jarrar, Mohammed Khalilia, and Sana Ghanem. 2022. [Wojood: Nested arabic named entity corpus and recognition using bert](#). In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2022)*, Marseille, France.
- Mustafa Jarrar, Sanad Malaysha, Tymaa Hammouda, and Mohammad Khalilia. 2023a. [Salma: Arabic sense-annotated corpus and wsd benchmarks](#). In *Proceedings of the 1st Arabic Natural Language Processing Conference (ArabicNLP), Part of the EMNLP 2023*. ACL.

- Mustafa Jarrar, Fadi Zaraket, Tymaa Hammouda, Daanish Masood Alavi, and Martin Waahlich. 2023b. [Lisan: Yemeni, irqi, libyan, and sudanese arabic dialect copora with morphological annotations](#). In *The 20th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA)*. IEEE.
- Alaa Joukhadar, Huda Saghergy, Leen Kweider, and Nada Ghneim. 2019. Arabic dialogue act recognition for textual chatbot systems. In *Proceedings of The First International Workshop on NLP Solutions for Under Resourced Languages (NSURL 2019) co-located with ICNLSP 2019-Short Papers*, pages 43–49.
- Kathrein Abu Kwaik, Motaz Saad, Stergios Chatzikyriakidis, and Simon Dobnik. 2018. [A lexical distance study of arabic dialects](#). *Procedia Computer Science*, 142:2–13. Arabic Computational Linguistics.
- Wuwei Lan, Yang Chen, Wei Xu, and Alan Ritter. 2020. Gigabert: Zero-shot transfer learning from english to arabic. In *Proceedings of The 2020 Conference on Empirical Methods on Natural Language Processing (EMNLP)*.
- Haneen Liqreina, Mustafa Jarrar, Mohammed Khalilia, Ahmed Oumar El-Shangiti, and Muhammad Abdul-Mageed. 2023. [Arabic fine-grained entity recognition](#). In *Proceedings of the 1st Arabic Natural Language Processing Conference (ArabicNLP), Part of the EMNLP 2023*. ACL.
- Ridha Mezzi, Aymen Yahyaoui, Mohamed Wassim Krir, Wadii Boulila, and Anis Koubaa. 2022. [Mental health intent recognition for arabic-speaking patients using the mini international neuropsychiatric interview \(mini\) and bert model](#). *Sensors*, 22(3).
- Eman Naser-Karajah, Nabil Arman, and Mustafa Jarrar. 2021. [Current trends and approaches in synonyms extraction: Potential adaptation to arabic](#). In *Proceedings of the 2021 International Conference on Information Technology (ICIT)*, pages 428–434, Amman, Jordan. IEEE.
- Amal Nayouf, Mustafa Jarrar, Fadi zaraket, Tymaa Hammouda, and Mohamad-Bassam Kurdy. 2023. [Nābra: Syrian arabic dialects with morphological annotations](#). In *Proceedings of the 1st Arabic Natural Language Processing Conference (ArabicNLP), Part of the EMNLP 2023*. ACL.
- Lorenz Nigst, Maxim Romanov, Sarah Bowen Savant, Masoumeh Seydi, and Peter Verkinderen. 2020. Openiti: a machine-readable corpus of islamicate texts. [nd http://doi.org/10.5281/zenodo, 4075046](http://doi.org/10.5281/zenodo.4075046).
- Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. [Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures](#). In *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*, Cardiff, United Kingdom. Leibniz-Institut für Deutsche Sprache.
- Robert Parker, David Graff, Ke Chen, Junbo Kong, and Kazuaki Maeda. 2011. [Arabic gigaword fifth edition ldc2011t11](#). *Philadelphia: Linguistic Data Consortium*.
- Sana Shams and Muhammad Aslam. 2022. [Improving user intent detection in urdu web queries with capsule net architectures](#). *Applied Sciences*, 12(22).
- Sana Shams, Muhammad Aslam, and Ana Maria Martinez-Enriquez. 2019. Lexical intent recognition in urdu queries using deep neural networks. In *Advances in Soft Computing*, pages 39–50, Cham. Springer International Publishing.
- Imad Zeroual, Dirk Goldhahn, Thomas Eckart, and Abdelhak Lakhouaja. 2019. [OSIAN: Open source international Arabic news corpus - preparation and integration into the CLARIN-infrastructure](#). In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 175–182, Florence, Italy. Association for Computational Linguistics.
- Yunhua Zhou, Peiju Liu, and Xipeng Qiu. 2022. [Knn-contrastive learning for out-of-domain intent classification](#). pages 5129–5141. Association for Computational Linguistics.

## A Error Simulation Types

### A.1 Common Errors ( $sim_c$ )

$sim_c$  are common spelling errors and word variations that people often make in real-life, which we derive from a lexicon. In a previous work, we developed a lexicon that contains a list of base forms, and the lexical variants (mostly colloquial terms) of each base form. The lexicon curation process started by collecting data from social media sites, chatbots and call centers audio recordings, which were transcribed manually. For each lexical variant, colloquial term and misspelling, the goal was to find its corresponding base form. Hence, a base form in the lexicon can have more than one lexical variant. The lexicon contains 12,111 base forms. To simulate these errors in our intent detection task, for each query, we randomly selected one to two words that have a matching base form in the lexicon, and for each base form we randomly selected one of its lexical variants. Because these errors are not simulated and are mostly colloquial variants collected from real content, we injected this type of error into the test set only, which will give us an insight how robust the model’s performance is on such noisy data. Examples of orthographic variants are shown in Table 9. For instance, the world شكرًا/thanks has four variants (شكرا, ششكرا, شكرًا, and شكرًا).

Lexical Term	Lexical Variants
شكرا	شكرا ششكرا شكرا شككرا
مثلا	مثلاث مثل

Table 9: Sample of lexicon, some words are colloquial while others are misspellings.

## A.2 Simulated Errors ( $sim_s$ )

$sim_s$  are errors simulated by deleting spaces between words. We applied this type of simulation on both the train and test sets. For each query we randomly deleted one or two spaces.

## A.3 Keyboard Errors ( $sim_k$ )

$sim_k$  are errors generated by inserting or deleting a letter from a word, replacing a letter with another letter, or swapping the places of two adjacent letters. Two approaches we followed when simulating this error. Either random replacement or replacement guided by the keyboard layout of the target language. Keyboard layout guided simulation will delete/insert/replace/swap letters based on the neighboring letters on the keyboard.