Evaluating NLP Models on Arabic Dialects: A Comparative Study of Encoder Performance and LLM Capabilities

Huang Baiyu
School of Mathematics Science
Peking University
2100010860@stu.pku.edu.cn

Shen Jingqi
School of Foreign Language
Peking University
2200018618@stu.pku.edu.cn

Abstract

With the development of Natural Language Processing (NLP), we have witnessed the great power of Large Language Models (LLMs). However, most of NLP technologies are based on English material. Researches focused on other languages, especially in dialects, are much lesser. Our work aims to evaluate how will the source of language affects the performance of a model. We will take Arabic Dialects as examples and conduct two aspects of experiments, including evaluating different encoders and examining LLMs to see whether they can understand dialects as well as Modern Standard Arabic.

1 Introduction

1.1 Background

Under the guidance of the *Pretraining-Finetuning* paradigm, the field of Natural Language Processing (NLP) has achieved remarkable successes with models such as BERT and ChatGPT. However, most of this work focuses on processing high-resource languages such as English. After convincing the success of ChatGPT, training models from other high-resource language from scratch is no doubt worthy.

When it comes to Arabic dialects, is it necessary to conduct specific training for these dialects, or are the existing models sufficient to address the challenges posed by dialectal variations? We should also point out the fact that, there are many kinds of Arabic Dialects widely distributed from North Africa to the Middle East. It would be expensive if we separately train models for each dialect. We need to make judgements on *Performance-Expense* payoff to solve the problems derived from dialects.

Meanwhile, after reading related works, we noticed that most of researches are paying effort to recognize dialects from standard language. We need to consider that if we really need to distinguish dialects, to make language models understand and deal with problems, since the modern *End-to-End* methods has greatly exceeded the classical NLP methods.

From a linguistic perspective, in Arabic, the phenomenon of "diglossia" is widespread, meaning that in addition to Modern Standard Arabic (MSA), people in different Arab regions also use their local Arabic dialects. Standard Arabic evolved from Classical Arabic and is primarily used in written language and formal occasions, largely based on the Quran. Therefore, MSA has fixed grammatical rules and pronunciation standards that are uniform across the Arab world. Meanwhile, in daily life, people usually use regional dialects, which can vary significantly due to varying influences from other languages and cultures (Turkish, French, Persian). Broadly, these dialects can be divided into five regions: Egyptian dialect, Levantine dialect (Sham dialect), Iraqi dialect, Gulf dialect, and North African (Maghreb) dialect. Among the many Arabic dialects, the Egyptian dialect is widely known and understood across the Arab world, thanks to the thriving film industry in Egypt of the last century, making it representative.Therefore, we evaluated and compared the LLM using the Egyptian dialect.

1.2 Contribution

Our contribution is listed as below:

1. We conducted experiments on different tokenizers. Our results prove that models trained on similar language will do help to understand dialects.

2. We also evaluated LLMs' ability to deal with dialects. Our results prove that it is not necessary to train a model specifically for a certain language, and good results can be achieved by adjusting the existing model.

Our project can be seen in Github and also submitted to the Course Web.

2 Methodology

2.1 Task Description

The language we are studying is Arabic and its dialects, including Moroccan Arabic, Algeria Arabic and Egyptian Arabic. We conducted two aspects of experiments, to evaluate whether the NLP models can handle dialects well.

In fundamental aspect, we want to evaluate how will the tokenizer affects the performance, given totally same models. To achieve this idea, we conducted two subtasks. One of them is the SemEval2024-Semantic Relateness(Ousidhoum et al., 2024a), which provides labelled dialect datasets. We also conducted experiments on Arabic Sentiment Analysis (SA), which is a more mature and classic task, with a large number of readily available datasets. Even though there's no dialect labels in SA datasets, we can use dialect recognition models to automatically label them. We will discuss them in detail in section 2.2.

From the point of view of a mature product, we also evaluated how will LLMs perform facing with dialectal inputs. We have written some test cases in Egyptian dialects to evaluate the LLMs' ability on understanding dialectal instructions and generating dialect-style contents. After obtaining the model's responses, we evaluated LLMs perform from the perspective of dialect word recognition and paragraph, and we presented them to native speakers to assess the quality of the responses.

2.2 Dataset

Generally speaking, the text dataset we provided to LLMs, is also kind of dataset. However, the datasets we mentioned here are specified to which are used for model training and evaluating.

As is mentioned before, we used the dataset from SemEval 2024 sub-task A(Ousidhoum et al., 2024b), to accomplish the Semantic Relatedness (STR) task. This dataset includes many kinds of languages and we selected two from them, namely Moroccan Arabic and Algerian Arabic. The Algerian Arabic dataset contains 1262 train samples and 584 test samples. A dataset for developing is also provided. The Moroccan dataset contains 925 train samples and 427 test samples. It is reasonable that a dialectal dataset doesn't contain so much data, and it's enough for fine-tuning a pretrained model. Each data contains a PairID, the unique identifier of each data, raw text separated by a line break and this sentence pair's relatedness score. An example is displayed as Table 1.

PairID	Text	Score
ARQ-train-0001	...	0.50

Table 1: An example of STR dataset

As for the Sentiment Analysis (SA) track, we also used a dataset from SemEval in 2018(Mohammad et al., 2018). It offers an Arabic Tweeter (Now called X) dataset and contains 2279 samples for train and 1519 samples for test. Each data in the SA dataset has many attributes, including ID, Tweet content, 11 types of sentiment. The 11 sentiments are not exclusive so each sentence may have many sentiment labels. Therefore we indeed need to do 11 binary classification for each sentiment.

2.3 Evaluate Encoder

We used four encoder, which are trained on different language and trained by different organization, so that they are likely to have different performance.

We used *bert-base-uncased* (Devlin et al., 2018) encoder as the baseline. As it is the first and famous pretrain encoder, it has been proved to have a strong ability. However, according to its paper, this model was trained on mainly English corpus, such as English Wikipedia and English books. We would regard it as baseline and compare it with those models trained on Arabic material.

Before we dive into Arabic, we also took multilingual BERT into consideration. Google has also trained such a model, namely *bert-base-multilingual-cased* (Devlin et al., 2018). In fact this model was also proposed in the same paper as bert-base. This model was trained on more than one hundred languages and doubtlessly containing Arabic. We hope to figure out do we need to specially train an Arabic model, or just train a model that knows every language.

The Arabic encoder we used are *bert-base-arabic* (Safaya et al., 2020) and *arabertv02* (Antoun et al., 2020). The first model was attractive that their corpus and vocabulary set are not restricted to Modern Standard Arabic. They contain some dialectical Arabic too, which is perfectly fitting with our task. The second encoder is outstanding on its large scale of training material, covering a large range of Arabic country including Egypt, Lebanon and of course Algeria and Morocco.

We built models that only differ on encoders, so that we can judge their performance simply by comparing their results, no matter how the model is. In other words, we don't need to tune the model for best performance.

2.4 Evaluate LLM

As previously mentioned, the Egyptian dialect is one of the most representative dialects in Arabic. Therefore, we used the Egyptian dialect to evaluate the ability of three large language models to handle dialects. These three models are Chat-GPT4o, primarily trained with English data; Chat-GLM4, primarily trained with English and Chinese data; and Jais, primarily trained with Arabic and English data. We compared the specific performance differences among the three models.

Our evaluation of the LLMs focused on two aspects: understanding dialect texts and generating dialect texts.

Regarding the understanding of dialect texts, since Arabic dialects are mainly used in daily life, we selected five complete dialogues in the Egyptian dialect and tasked the three models with direct translation and summarizing the content to assess their understanding of the texts. We examined two aspects of the translation results:

Recognition of Dialectal Words: In the original text, we had already marked words/phrases with dialectal features. By examining the translation results, we could see which dialectal words the models understood. Based on this, we compiled statistics.

Discourse-Level Evaluation: Observations revealed that merely identifying dialectal words was insufficient to assess the models' understanding of the text. This is because the generated results might not be direct translations but rather modified outputs with additions or omissions, potentially altering the meaning entirely. Therefore, we needed to consider the semantic completeness and coherence of the translation results from a discourse-level perspective. Thus, we scored the overall semantic completeness and accuracy of the translations.

For the task of summarizing the content, since we mainly evaluate the degree of understanding of LLMs, we mainly examine the completeness and correctness of the summary. We gave the results generated by LLMs to native speakers for scoring, and also put the results generated by LLMs and their corresponding translations in the attached files for easy comparison with the Chinese translation of the original text.

To generate dialect texts, we will directly use instructions in the Egyptian dialect to ask three major models to generate the following five types of texts in Egyptian dialect: weather forecast, news report, short story, dialogue, and advertisement. These texts have a relatively strong colloquial flavor, which aligns with the characteristic that dialects are mainly used in spoken language. Subsequently, we will have native speakers evaluate the generated texts based on two main criteria: whether the style leans more towards the Egyptian dialect or standard language and the accuracy of the Egyptian dialect used.

3 Experiments

3.1 Enviroment

Our programming language is Python, in the version of 3.10. Based on PyTorch framework, we build several deep learning models for our tasks. We use personal computer to do all calculation. Our GPU device is NVIDIA GeForce RTX 3060 Laptop GPU, which can greatly accelerate our training procedure.

3.2 Implementation Details

3.2.1 Data Preprocess

The SemEval 2018 dataset has not location label. Since our goal is to do experiments on Arabic dialects, we need to distinguish dialects from Modern Standard Arabic.

We use a ready-made classification model from GitHub[1] to label all sentence. This model can classify each sentence into an Arabic country, such as Yemen, Eygpt and so on. To align with the STR task, we choose those sentence labelled as Morocco and Algeria A bar figure1 is shown. From that figure we can see, Morocco and Algeria data has a visible large amount, comparing to Iraq and Syria. With such a sufficient scale of data amount, this also indicates that the two language are good dialects to research.

Even though we have labelled every sentence, we still use all data in train set to train our model. We only use these dialectal data for test.

[1] https://github.com/Lafifi-24/arabic-dialect-identification

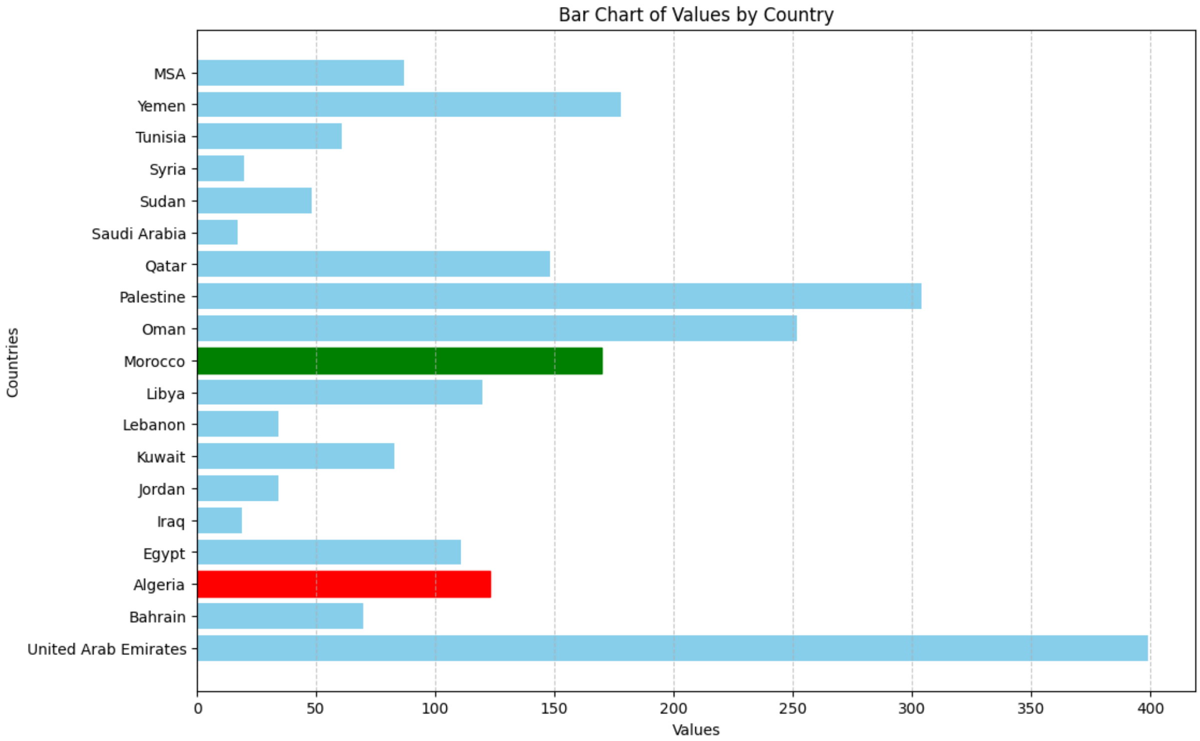


Figure 1: Country distribution in Arabic SA Dataset. The two languages we selected are in different color.

3.2.2 Loss Function

The SA task's loss function is no doubt to be BCELoss, namely binary cross entropy, which is very suitable for the binary classification task. However, the STR task's loss function is not so easy to select.

At the first glance we may try MSE regarding it as a regression task. However, after checking the evaluation model provided by organizer, we noticed that they calculate the Spearman's Rank Order Correlation (Spearmanr)(Spearman, 1904) score as the evaluation metric, which is used to measure the trend of two series. That is to say we don't need to get the same digital value as labels, but only general trend.

We may consider using this Spearmanr as our loss. However, as a score calculating two series' rank order trend, this function is discrete and not derivable. This problem is fatal since deep learning model needs a derivable loss function to back propagation.

After doing some research, we decided to use Pearson's Correlation(Pearson and for National Eugenics, 1895) to estimate Spearmanr. The Pearson Correlation is defined to measure two series' linear correlation, and it can represent the Spearmanr in some degree. What's more important is that this function is derivable, ensuring it to be a loss function.

After selected this function, we can implement it in PyTorch. We also tried using MSE as loss function but turned out to be a bad choice. It's not surprising because we selected a bad optimization goal. However, this problem also reveals the fact that this task was not very mature and well-organized, and likely to get a low result score.

3.2.3 Questions for LLMs

As mentioned above, the questions we asked LLMs mainly focused on two aspects, respectively examining LLMs' understanding and production abilities of dialects. We selected the more representative Egyptian dialect as the dialect for LLMs' dialect processing ability assessment.

In the comprehension ability assessment, we selected five daily Egyptian dialect dialogues with life as the test texts, and asked LLMs to perform the following two operations on each text: 1. Translate the text into Chinese/English (Jais does not support Chinese) 2. Summarize the content of the text The above questions were all asked in the Egyptian dialect.

In the production ability assessment, we asked LLMs to generate five different texts with strong spoken colors in the Egyptian dialect. These five texts are: weather forecast; news report; short story; dialogue; advertisement. The above questions were also asked in the Egyptian dialect.

4 Results

4.1 STR task

As we claimed in section 3.2.2, we didn't achieve a high score. All of our results are displayed in below Table 2. The loss mentioned here is training loss after 5 epochs.

We have to admit that all these scores are low. However, the results displayed in SemEval 2024 have same phenomenon. The team who gets highest score in some language also gets a negative score in other language.

Back to our results, we noticed that Arabic models converge faster that the other two, and reached a higher score.

model	loss	spearmanr
bert-base	0.901	-0.040
multilingual-bert	0.905	-0.046
bert-base-arabic	0.836	0.028
arabert	0.769	0.009

Table 2: STR results

4.2 SA task

As a much more mature task, we got results that are visible and convincing. The loss function's value in 10 epochs is shown in Figure 2. As we can see in this figure, the three models except baseline model, converge fast and reach a low loss value. Finally the multilingual BERT model reached the lowest loss, and the two Arabic models are slightly higher than it.

On test set, we use these four models to see if they can correctly classify sentences. The results can be seen in Figure 3. The blue bar and orange bar are Arabic models. The y axis indicates the number that models correctly classified. It starts from 120 to emphasise each model's difference. Even the baseline, also reached a high accuracy.

However, in most cases Arabic models reached a higher accuracy. Sometimes multilingual models can come up with Arabic models but the difference is not significant.

We can also see that *bert-base-arabic* model always performs better than *arabert*. We will discuss this phenomenon later.

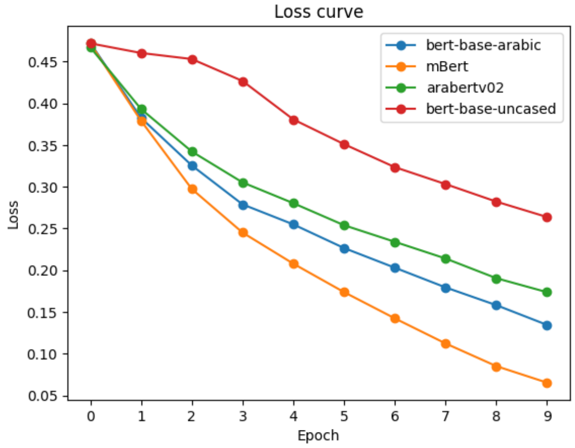


Figure 2: Enter Caption

4.3 LLM QA

We evaluated the comprehension and generation capabilities of LLMs and found the following results .The statistical results of LLMs for dialect word recognition are in Table 3.

In the scoring shown in Figure[2] 4, the main considerations for completeness scoring are: whether there are omissions in the translation, whether there are any loss of meaning, and whether the paragraph is complete; while the main considerations for accuracy scoring are: whether the translation is accurate, whether there are excessive deletions or additions, and whether it deviates from the original meaning.

We also asked native speakers for rating LLM responses, results are displayed in Figure 5. Similar to the previous score, the main considerations for the completeness score in this score are: whether there are omissions in the summary, whether there are any missing meanings, and the degree of information coverage; while the main considerations for the accuracy score are: whether the summary is accurate, whether there are excessive deletions or additions, and whether it deviates from the original meaning.

Finally the rating of he text generated by LLMs, is presented in Figure 6.In this scoring, the main considerations for the dialectalization score are: whether the style is more inclined to standard Arabic or Egyptian dialect, and whether it is more in line with the expression habits of standard Arabic or dialect; while the main considerations for the dialect usage accuracy score are: whether the dialect used is appropriate, and whether the dialect used is Egyptian dialect rather than other dialects.

[2] We insert a picture instead of table for convenience.

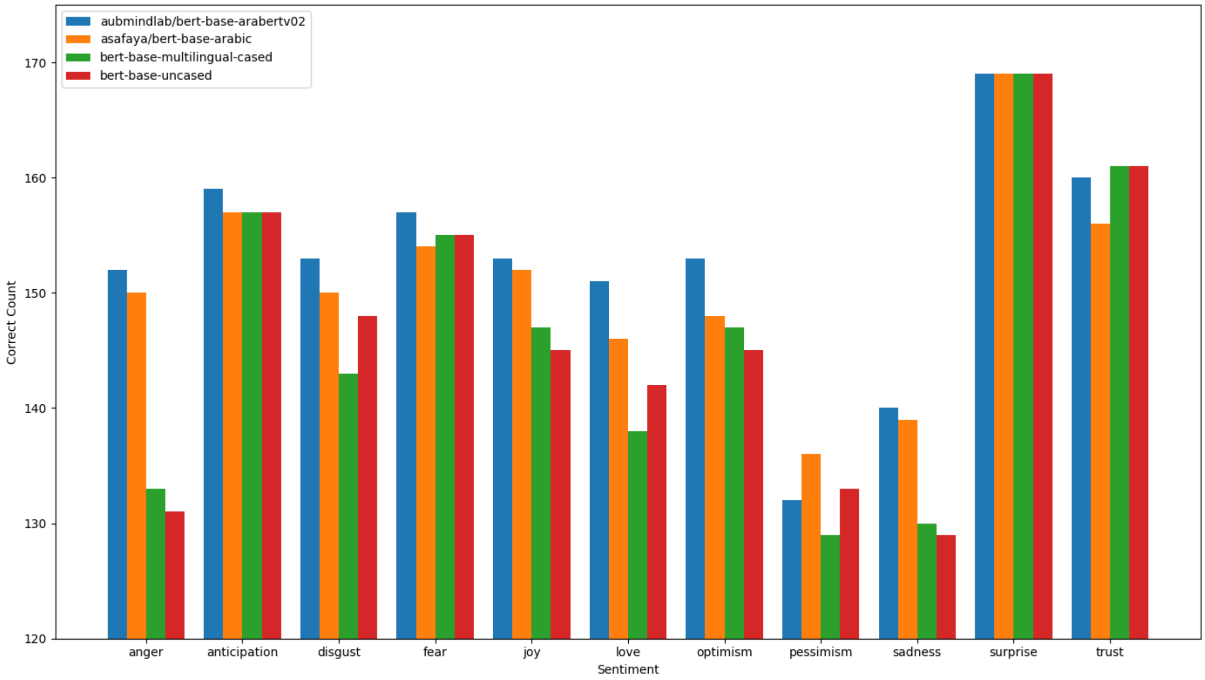


Figure 3: Model performance in each sentiment

5 Analysis

5.1 Encoders

From the subtask that evaluating encoders, we can say that Arabic models have a better performance than multilingual models.

We can find explanation from their training materials. Provided with so much Arabic text material, these models should have a better performance on solving Arabic problems. From Figure 3 we can see, the model with largest Arabic training material, represented by the blue bar in the chart, achieved the best performance. The *bert-base-arabic* model, represented by orange bar, takes smaller size dataset to train and gets a slightly lower score.

This can also be explained after reading their origin paper. As one of the first pre-trained models proposed by Google, *bert-base-arabic* simply replaced the language from English to Arabic and was trained on Arabic corpus. In contrast, *arabert* was proposed in 2020, building on the experience of previous explorations so it can do better.

In addition, according to the paper (Antoun et al., 2020), *arabert* also made optimizations specific to the Arabic language. Arabic is known for its lexical sparsity, so they segmented Arabic words into into stems, prefixes and suffixes. Then they trained model by predicting each sub-word individually, as what they do in BERT.

By comparing these four models comprehensively, we propose that the amount of specific language text used to train a model significantly affects the model's ability to process that language.

The *bert-base-uncased* is doubtlessly containing least amount of Arabic text, and it reached the lowest score in our experiments. But it also trained a lot and has the ability to do as much as it can. So it still got a not bad performance in SA task. The *multilingual-bert-cased* contains some Arabic material and other languages' material, enabling it to deal with Arabic tasks. But the amount of language material is relatively small, so it didn't perform as well as Arabic models.

The *bert-base-arabic* model is based on BERT framework and uses specified Arabic dataset for training. The *arabert* model takes larger Arabic datasets into consideration and makes some optimization for Arabic, and therefore performs best.

From the above discussion, we can conclude that using more relevant corpora for training can enhance a model's performance. If there is a need to target Arabic dialects, providing more dialect-specific materials can improve the model's ability to handle those dialects.

5.2 LLM

However, in Section 4.3, we examined the performance of large-scale models and found that large models Jais, which is specialized in Arabic perform worse compared to ChatGPT, which is trained on a broad dataset. Even ChatGLM can do better than Jais.

We can notice that Jais stops updating their model since 2023.8 [3], yet OpenAI has consistently continued to update its models after publishing the world-shaking ChatGPT. ChatGLM is also continuously updating its model[4]. We must consider whether Jais's poor performance is due to its lack of continued model iterations.

In the evaluation of the ability of large models to understand dialects, from the vocabulary level, Chat-GPT performed best in all texts. In the relatively worst understood text 1, the parts that Chat-GPT could not understand were all colloquial words with strong Egyptian dialect characteristics, and the other two models did not understand these words. From the vocabulary point of view alone, the number of Egyptian dialect words that Chat-GLM4 and Jais models can translate is basically the same, but from the comprehensive text, we found that due to the large difference between the expression of numbers in Egyptian dialect and standard Arabic, Chat-GLM4's processing effect on numbers in Egyptian dialect is very unsatisfactory, with an accuracy rate of less than 10%. Although Jais also has many errors in understanding numbers, its accuracy rate is higher than Chat-GLM4. On the other hand, although the number of Egyptian dialect words that Chat-GLM4 and Jais models can translate is basically the same, it does not mean that the translation quality is similar. In fact, from the perspective of discourse, Chat-GLM4 has significantly higher translation quality and better understanding of the original text.

From the perspective of discourse, it is not difficult to find that the translation given by Chat-GPT is basically consistent with the manual translation. The translation given by Chat-GPT is complete and accurate, basically corresponding to the original text, without deletion or addition of content, and the expression is fluent. As for the inaccurate translation of Chat-GPT, it is also excusable: in the Chat-GPT translation text of Text 1, the translation of "the best person" in the third and fourth lines is inaccurate. This is only the surface meaning of the Arabic original text, but in fact it is a verbal expression similar to an allusion. This sentence actually praises the place mentioned by praising people. Therefore, in the Egyptian dialect, this sentence usually expresses the meaning of "good place" in a fixed way; there is another inaccuracy in the Chat-GPT translation text of Text 1 because Chat-GPT did not understand the word "camel" in the Egyptian dialect and translated it into "matching sentences". In contrast, Chat-GLM4 understood that this word means "camel". As for the other two models, Chat-GLM4 and Jais, Chat-GLM4 performs much better than Jais overall, which is mainly reflected in the completeness and rationality of semantics. Although Chat-GLM4, as mentioned above, does not understand the meaning of some dialect words very well, Chat-GLM4 will try to complete the text according to the context, so that the translation results show relatively good completeness. However, there are many very obvious problems in the text processed by Jais, such as: ignoring the entire line of text, synthesizing two or even more lines of dialogues of different characters into one sentence, and excessive supplementation leading to a complete deviation from the original meaning... By comparing the manual translation of the dialogue with the translations generated by Chat-GLM4 and Jais, it can be found that Chat-GLM4's processing effect at the paragraph level is significantly better than Jais. In addition, Jais basically only accepts Arabic instructions, and its ability to understand instructions is significantly poorer than the other two models.

In the summarization task, it is not difficult to find from the data that Chat-GPT still performs best. The generated content is fluent, the summary is complete, the details are appropriate (for example, a large number-based conversation in the middle is summarized as "bargaining"), and the understanding is very accurate. Among the remaining two LLMs, Chat-GLM4 is better than Jais in overall performance and has better stability. The main problem of Chat-GLM4 is that it does not understand many dialect words or phrases (especially the usage of numbers), which leads to low translation accuracy, which is consistent with our analysis above, but Chat-GLM4's summary of the text is still relatively complete. Jais's performance is relatively poor. Not only does it miss many information points, but the overall meaning and the original meaning of the article often deviate to varying degrees. Therefore, overall, the text summarized by Jais has the lowest scores in completeness and accuracy.

Finally, in the attempt to evaluate the ability of

[3] Jais's Blog:https://inceptioniai.org/blogs/

[4] Zhipu News:https://www.zhipuai.cn/news

LLMs to generate dialects, we found, not surprisingly, that Chat-GPT is still the best performing LLM. This is reflected in the fact that it can not only understand the meaning of the instructions very well, but also the generated texts have strong dialect characteristics, and the use of these dialects is often not a big problem. The performance of the other two models is obviously much worse. Chat-GLM4 and Jais sometimes generate texts that do not reflect obvious dialect characteristics. This may also be related to the task. For example, in the generation of news reports, Chat-GLM4 generates very formal manuscripts, and these manuscripts are almost all written in standard Arabic, but the other two models generate more oral news reports, which can well reflect the characteristics of dialects. In addition, Jais's ability to understand instructions is obviously inferior to the other two models, and sometimes it takes repeated attempts to generate the required content.

6 Conclusion

In summary, this paper investigates the performance of different text encoders when processing Arabic dialects. The comprehensive comparison concludes that models primarily trained on Arabic data perform better than those not specifically tailored for Arabic. This indicates that if further optimization for Arabic dialect issues is required, increasing the quantity and quality of data is an essential step.

On the other hand, among the existing large models, the large models specifically for Arabic are not better than the large models mainly trained with English or other languages in dealing with Arabic dialects. In fact, judging from the performance of Chat-GPT4o, its ability to handle Arabic dialects has reached a very good level. From this point of view, we do not need to train models for different languages separately, and we can achieve better results through appropriate adjustments.

References

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Saif M. Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 Task 1: Affect in tweets. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, USA.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 13 languages. In *Findings of the Association for Computational Linguistics: ACL 2024*. Association for Computational Linguistics.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

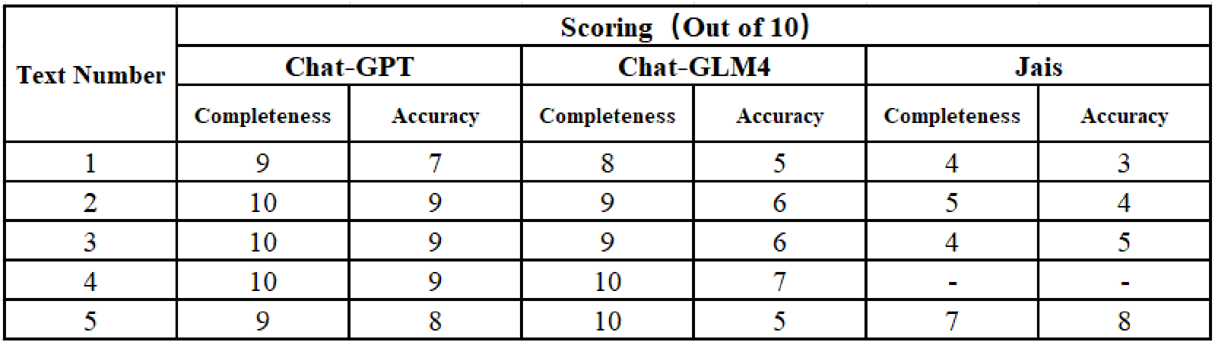
K. Pearson and Galton Laboratory for National Eugenics. 1895. *"Note on Regression and Inheritance in the Case of Two Parents"*. Proceedings of the Royal Society. Royal Society.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059, Barcelona (online). International Committee for Computational Linguistics.

C. Spearman. 1904. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101.

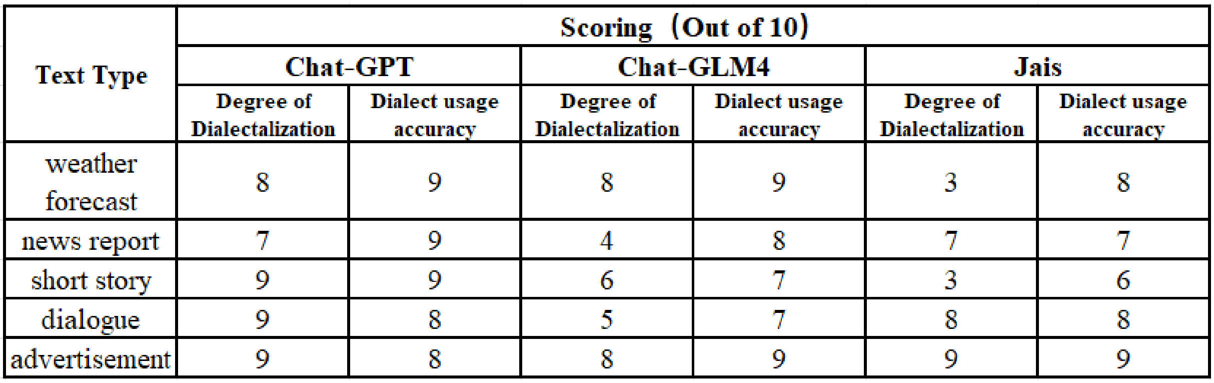
Text ID	Number of Dialect words	ChatGPT	ChatGLM4	Jais
1	36	30	25	28
2	43	43	38	41
3	32	32	29	25
4	37	37	32	-
5	41	40	37	36

Table 3: Number of words understood by the large model. Data is missing at - in the table. Text 4 is a daily conversation about food, and does not contain sensitive content. However, Jais repeatedly indicates that the content policy is violated.



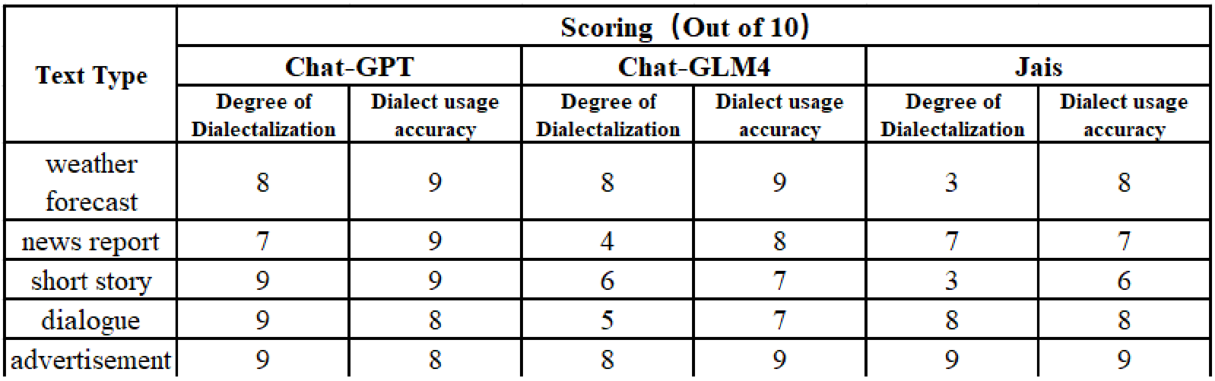
Text Number	Scoring（Out of 10）					
	Chat-GPT		Chat-GLM4		Jais	
	Completeness	Accuracy	Completeness	Accuracy	Completeness	Accuracy
1	9	7	8	5	4	3
2	10	9	9	6	5	4
3	10	9	9	6	4	5
4	10	9	10	7	-	-
5	9	8	10	5	7	8

Figure 4: Ratings of translation results at the discourse level (out of 10).
Data is missing at - in the table. Text 4 is a daily conversation about food, and does not contain sensitive content. However, Jais repeatedly indicates that the content policy is violated.



Text Type	Scoring（Out of 10）					
	Chat-GPT		Chat-GLM4		Jais	
	Degree of Dialectalization	Dialect usage accuracy	Degree of Dialectalization	Dialect usage accuracy	Degree of Dialectalization	Dialect usage accuracy
weather forecast	8	9	8	9	3	8
news report	7	9	4	8	7	7
short story	9	9	6	7	3	6
dialogue	9	8	5	7	8	8
advertisement	9	8	8	9	9	9

Figure 5: Native speakers‘ rating of the LLMs’ summarization of the text (out of 10)



Text Type	Scoring（Out of 10）					
	Chat-GPT		Chat-GLM4		Jais	
	Degree of Dialectalization	Dialect usage accuracy	Degree of Dialectalization	Dialect usage accuracy	Degree of Dialectalization	Dialect usage accuracy
weather forecast	8	9	8	9	3	8
news report	7	9	4	8	7	7
short story	9	9	6	7	3	6
dialogue	9	8	5	7	8	8
advertisement	9	8	8	9	9	9

Figure 6: Native speakers‘ rating of the textes generated by LLMs