FROM LATIN TO ARABIC: INNOVATIVE APPROACHES FOR RETRIEVING ORIGINAL TEXTS AND VECTORIZING FRANCO-ARABIC

Mohamed Abd-Elnabi I. I. Gabr
Computer and Systems Engineering Department, Faculty of Engineering, Ain Shams University, Cairo, Egypt.
1902186@eng.asu.edu.eg

Ahmed Z. Badr
Computer and Systems Engineering Department, Faculty of Engineering, Ain Shams University, Cairo, Egypt.
ahmed.z.badr@eng.asu.edu.eg

Hani M. K. Mahdi
Computer and Systems Engineering Department, Faculty of Engineering, Ain Shams University, Cairo, Egypt.
hani.mahdi@eng.asu.edu.eg

Abstract

In the last few years, the importance of artificial intelligence, deep learning techniques, and transformer-based models in the analysis and understanding of English texts has emerged. It highlights this importance in various tasks, such as answering questions in smart chat systems, sentimental analysis, named entity recognition, and opinion polls analysis in various fields. The significant improvement in Internet services and their spread around the world caused a significant increase in the number of Internet users who speak Arabic. Interest in analyzing Arabic texts on various platforms has begun. This task posed a great challenge due to the difficulty of the Arabic language, its morphological richness, and the multiplicity of its dialects. In addition to the emergence of a new challenge, where many Arab users adopted writing the Arabic language using Latin letters due to the lack of support for the Arabic language at the beginning of the spread of the Internet. This way of writing is called the Franco-Arabic language or Arabizi. It is not English language nor Arabic language. We can call this process transliteration which concerns similar-sounding characters of another alphabet. Transliteration isn't always an exact science because sometimes words can be transliterated in more than one way. This caused uncertainty about converting Franco-Arabic to Arabic because there are many to many relationships between some Latin and Arabic characters in conversion based on the phonetic tone in addition to the multiplicity of writing methods from one Arab country to another. In this paper, we will introduce two methods for dealing with Franco-Arabic. The first method is concerned with retrieving original Arabic text based on pre-trained transformers-based models and the second method is concerned with training a new model that can vectorize and understand Egyptian Franco-Arabic texts directly.

***Keywords*: Machine learning; Deep learning; Natural Language Processing (NLP); Social Media; Arabizi.**

1. Introduction

There are 5.07 billion internet users and 4.74 billion active social media users in 2022 [1]. With the great development in Internet services and the spread of electronic services and social networking sites, there is a lot of Arabic content that requires analysis. Arab internet users grew to 237 million users grew by 9.3% from 2000 to 2020 and the Arabic language became number four among the top ten languages used on the web [2].

Written Arabic in social media can be divided into two types of texts. The first text type has only Arabic characters, and the other type is written in Latin characters and numbers. The last is known as Franco-Arabic. Arabic written contents can be divided into three types Classical Arabic, Modern Standard Arabic (MSA), and Colloquial Arabic [3]. Classical Arabic is the dialect that was spoken before and during the Islamic era, and it is the dialect in which the Holy Quran was later composed. It has complicated grammar and a wide vocabulary. The classical type gave rise to the Modern Standard Arabic (MSA) which is currently the official tongue for literature, media, and education. It uses less complex grammar. Finally, colloquial Arabic is slang that varies between Arab nations and regions.

Colloquial Arabic is the language that people use to communicate with one another daily. When personal computers and mobile phones first became more widely available to regular users in the 1990s, the history of Franco-Arabic officially began. Only the Latin alphabet was available for communication on these phones at the time the Arabic alphabet wasn't an optional feature before more languages were added [4]. The Arab users adapted and devised a way to write the Arabic language using Latin letters, which caused the establishment of an innovative writing method called Franco-Arabic or Arabizi like "al7b a3my" which means in Arabic " **الحب أعمى**" and means in English "love is blind". Dealing with Colloquial Egyptian Franco-Arabic is the concern of this research paper.

There is no strict specific way to write in Franco-Arabic. As shown in Table 1, not every character is mapped to only one but there is uncertainty in converting some characters. For example, the character mentioned in record 28 may be mapped to many characters. The selection of the mapped character mostly depends on the phonetic tone of the spoken character in the word. The Franco-Arabic language spread and became essential among many persons; especially among the youth, on social media platforms. To elucidate the difficulty of the translation between Franco-Arabic to Arabic and vice versa Table 2,3,4 show types of relationships between Arabic and Latin characters and numbers.

Table 1. Lookup Table for Characters [5]

Serial	Arabic character	Latin character
1	ء أ ؤ إ ئ آ	2
2	ا	a e è
3	ب	b p
4	ت	t
5	ث	s th t
6	ج	j dj g
7	ح	7 h
8	خ	kh 5 7’
9	د	d
10	ذ	z th dh d’
11	ر	r
12	ز	z
13	س	s
14	ش	sh ch $ 4
15	ص	s 9
16	ض	d dh 9' D
17	ط	t 6 T
18	ظ	z th dh 6'
19	ع	3
20	غ	gh 3’8
21	ف	f v
22	ق	2 g q 9 8
23	ك	k g ch
24	ل	l
25	م	m
26	ن	n
27	هـ	h a e ah eh
28	و	w o ou oo u
29	ى ي	y i ee ei ai a é
30	ة	a e eh at et é

Table 2. One to One Relationships

Arabic character	Latin character
ل	l
م	m
ن	n
ع	3

Table 3. One to Many Relationships Between Latin And Arabic Letters

Latin character	Arabic character						
2	ء	أ	آ	إ	ق	ؤ	ئ
8	غ	ق					
9	ص	ق					
a	ا	ة	هـ	ى	ي		
d	د	ض					
e	ة	ه					
e'	ا	ة	هـ	ى	ي		
h	ح	هـ					
s	ث	س	ص				
t	ت	ث	ط				
z	ذ	ز	ظ				
ch	ش	ك					
dh	ذ	ض	ظ				
eh	ة	ه					
th	ث	ذ	ظ				

Table 4. One to Many Relationships Between Arabic And Latin Letters

Arabic character	Latin character						
ا	**a**	**e**	**e'**				
ب	**b**	**p**					
ث	**s**	**t**	**th**				
ج	**j**	**dj**	**g**				
ح	**7**	**h**					
خ	**7'**	**5**	**kh**				
ذ	**d'**	**z**	**dh**	**th**			
ش	**4**	**$**	**ch**	**sh**			
ص	**9**	**s**					
ض	**9'**	**D**	**d**	**dh**			
ط	**6**	**T**	**t**				
ظ	**6'**	**z**	**ch**	**dh**			
غ	**3'**	**8**	**gh**				
ق	**2**	**8**	**9**	**g**	**q**		
ك	**g**	**K**	**ch**				
ه	**a**	**e**	**e'**	**h**	**ah**	**eh**	
و	**o**	**u**	**w**	**oo**	**ou**		
ة	**a**	**e**	**e'**	**eh**	**at**	**et**	
ى, ي	**a**	**i**	**e'**	**y**	**ee**	**ei**	**ai**

Most of the previously done work in the fields of Natural Language Processing (NLP) such as sentiment analysis, mood extraction, named entity recognition, and others was developed to analyze English text. In addition, most of the available data sets are in English. Some work has been done to analyze Arabic text, but to the best knowledge of the authors, these are not targeted Franco-Arabic texts.

There are diverse ways to deal with text written in Franco-Arabic, each having its pros and cons [6]. There are three main approaches or strategies for processing with Franco-Arabic texts. The first approach is to convert Franco-Arabic to English and take advantage of the availability of English datasets and the NLP toolkits like NTLK [7]. The major drawback of this approach is translating Franco-Arabic to English goes through three phases. The first one converts Franco-Arabic to Arabic letters then converts Arabic which is written in Colloquial language to MSA. The last phase converts Arabic MSA to English. These phases may affect the meaning of the text or loss of feeling from the original content.

The second approach for dealing with the Franco-Arabic texts is to convert from Franco-Arabic to Arabic texts, whether the converted text will be in Modern Standard Arabic or Dialect Arabic. Since there is only one stage in the conversion, the words may still have the same pronunciation but different spelling. Of course, there is a percentage of errors due to the ambiguity of the conversion. Reducing this ambiguity is a concern to us because it is considered the foundation for completing the first and second strategies. So, it is considered the foundation for completing the work. The first concern in this paper is a method to reduce this ambiguity.

The third and final strategy uses original content that was written in Franco-Arabic. Since the words input by the users is the ones being processed, this method ensures that the sentiments and meanings of the original text are preserved. However, there are no datasets or lexicons that can be utilized for NLP tasks because, to our knowledge, extremely little to no work and research has been done on Franco-Arabic literature.

The inability of the pre-processing technologies to recognize or comprehend content written in Franco-Arabic is another disadvantage. An Arabic word written in any alphabet would be unrecognizable even by NLP toolkits designed to handle Arabic text other than the Arabic alphabet. To deal with these issues in the third strategy, we innovated a method that can convert any Arabic text to Franco-Arabic text. The resulting sentences dataset is then vectorized using a new model. By this model semantic similarity between Franco-Arabic sentences can be viewed and measured. The semantic similarity scores prove the ability of the new model to understand Franco-Arabic directly hence used in various NLP problems.

2. Related Work

In 2018, Magdy et al. [6] used the second strategy after some enhancement to extract mood from Franco-Arabic Facebook posts. In 2018, Soliman et al.[8] built a Slang Sentimental Words and Idioms Lexicon (SSWIL) of opinion words used by Arab youth in their comments on news topics on a variety of social network websites using Franco-Arabic. This step enhanced analyzing social network opinion mining for slang Arabic language. In 2023, Bousri et al. [9] proposed a system for Rumor Detection in Algerian Franco-Arabic. In 2022, Hajbi et al. [10] propose a new method based on NLP to address the challenges of pre-processing text that contains Franco-Arabic Based on Deep Learning and Associations. In 2021, Fourati et al. [11] introduce a large Tunisian Arabizi dialectal dataset for Sentiment Analysis. In 2021, Talafha et al. [12] introduce an attention-based Long Short-Term Memory model for Arabizi transliteration.

In NLP, text vectorization, also known as "vector embedding," is frequently utilized where words, phrases, or even larger units of text are represented as vectors. The first attempt to get the embedding of words was applied by Luhn [13]. It is Count-Based Text Vectorization dependent on counting unique word terms in a document disregarding the order of the words or the grammar and giving weight to every word corresponding to the frequency in the document. The term frequency of documents is not the best representation because common terms like “and, the, to” has usually the maximum frequency terms in the documents. So, the terms with the highest frequencies do not mean that those words are the most important terms in the document. A Bag of Words (BOW) is an algorithm that can apply the term frequency method to documents [14]. Term frequency will tend to incorrectly focus attention on documents, which happen to use the words like “and, from, to” more frequently, without giving enough weight to the more meaningful terms. Hence, an Inverse Document Frequency Factor IDF introduced by Sparck Jones [15] is incorporated which decreases the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely. TF-IDF is easy to compute and easily computes the similarity between 2 documents but it has drawbacks of disability of capture semantics.

Recently, improvements in neural network techniques combined with other innovative research such as attention mechanisms and transformers resulted in several novel innovations: Word2vec from Google [16], Glove from Stanford University [17], Fasttext from Facebook AI Research [18] and recently transformers-based models From Google research [19]. When Google researchers introduced Pre-training of Deep Bidirectional Transformers (BERT) for Language Understanding, a new era in Natural Language Processing (NLP) began [20]. The encoder-decoder transformer architecture is used by the BERT-based model to learn the semantic

similarity of the input questions mechanism. BERT is intended to jointly condition both left and right contexts in all layers to both left and right contexts in all layers to pre-train deep bidirectional representations from the unlabeled text. Therefore, state-of-the-art models for a variety of tasks, such as question answering and language inference, may be created using the pre-trained BERT model with just one additional output layer, without significant task-specific architecture adjustments [20].

3. Dataset Analysis and Setup Environment

In this section, the considered dataset in this paper and the main pre-processing techniques as well as the experimental environment are termed.

3.1. *Data collection*

As mentioned above there is a dataset to extract the Egyptian slang dictionaries and build Franco-Arabic samples. There are no datasets or lexicons for Franco-Arabic because truly little work and research have been done on Franco-Arabic literature. Because we are interested in Egyptian slang, so we used a new Arabic corpus for Egyptian tweets [21]. This corpus consists of forty thousand tweets. Additionally, we need to enlarge our dataset to cover all topics as possible, so we collected about 183612 records of Egyptian slang using Twitter API [22]. The overall shape of the dataset is 223612 covering a blend of different general topics discussed on Twitter. We innovated a method that will be discussed later, it can be used to transliterate Arabic texts to Franco-Arabic and vice versa. We used this method to convert a sample of the Egyptian Tweets datasets to Franco-Arabic. Also, we used these datasets to build a dictionary vocab for Egyptian Slang that contains 240219 unique vocabularies.

3.2. *Data preprocess*

Various Arabic pre-processing techniques were used to prepare the dataset for the training phase, improve accuracy, and minimize the noise in the data. These are:

- Elimination of non-Arabic words.
- Elimination of hashtags.
- Elimination of hyperlinks.
- Elimination of Arabic diacritics and elongation
- Elimination of punctuation and symbols such as “? (,),’ ! @ \$ \# —”.
- Normalization, which is used to remove “ء” from the “أإآ”.

Special functions are written in python and used to normalize Arabic letters, remove diacritics and Arabic punctuations.

3.3. *Setup environment*

The hardware that is used for training models is a server having (4) physical processors with specifications Intel® Xeon® Processor E7540 18M Cache, 2.00 GHz, 6.40 GT/s, 6 cores, and 12 threads, and running under the operating system Ubuntu version 20.04. We built a Conda environment using Python 3.9 with the required packages which are pandas, numpy, matplotlib, sklearn, TensorFlow, PyTorch, yamli, flask, jupyter notebook, transliterate, cameltools, spellchecker, enchant and transformers.

3.4. *Franco-arabic conversion and reducing errors*

There are six steps will be used to create a random sample dataset from the above-mentioned Egyptian Twitter. This sample will be used to check the performance of the conversion method as follows:

- Apply cleaning data to the original sentence before converting to Franco-Arabic.
- Convert Arabic to Franco-Arabic.
- Convert Franco-Arabic to Arabic sentence using Yamli, which will be discussed later.
- Exclude names and places then detect error words using a built dictionary.
- Replace every error word with the [MASK] flag and predict the correction word.
- Correct the masked words using the pre-trained model.

Every step will be illustrated in detail in the following subsections. We used a sample of 500 records from Egyptian twitter and apply the previous step to generate a dataset that contains original Arabic sentences and Arabic sentences that result after predicting MASKED words. Table 5 shows an example for the steps applied to the sentence.

TABLE 5. STEPS ON SENTENCE EXAMPLE

Original sent	من شب على شئ شاب عليه.
Original sent preprocess	من شب على شئ شاب علىه
Franco-Arabic	mn shb 3ly sh2 shab 3lyh
Yamli Arabic	من شب علي ش٤ شاب عليه
Masked sent	من شب علي [MASK] شاب عليه
Corrected sent	من شب علي شى شاب عليه

3.5. *Converting arabic to franco-arabic*

In this subsection, the first two steps are considered. As mentioned above, the first step concerns cleaning and normalizing the dataset to reduce ambiguity before conversion. As shown In Table 3 there is confusion with some letters like "ة"-"ه", "ي"-"ى", "ک"-"ك","ؤ"-"و" and "أإآ"-"ا". So, we normalized these characters to reduce confusion before converting them to Latin characters.

The second step is to build a Franco-Arabic dataset sample using our innovative method to convert Arabic to Franco-Arabic. We used the python bidirectional transliterate package [23]. Transliterates Unicode strings according to the rules specified in the language packs. The package comes with some language that does not include the Arabic language. So, we registered the Arabic alphabet. After registration a looping algorithm is applied to our sentence characters to pass it to the mapping equations. The first one is called the prep-processing mapping equation which is responsible to convert Arabic characters if found to their most used compound Latin characters like "th"-"ث". If a character is not founded in pre-processing mapping then we used a mapping equation to convert the character to single Latin letters. This process is bidirectional through a used library. the steps as shown in Fig1. We used this method to convert any Arabic dataset to Franco-Arabic which will help us to build a model able to understand Franco-Arabic Directly.

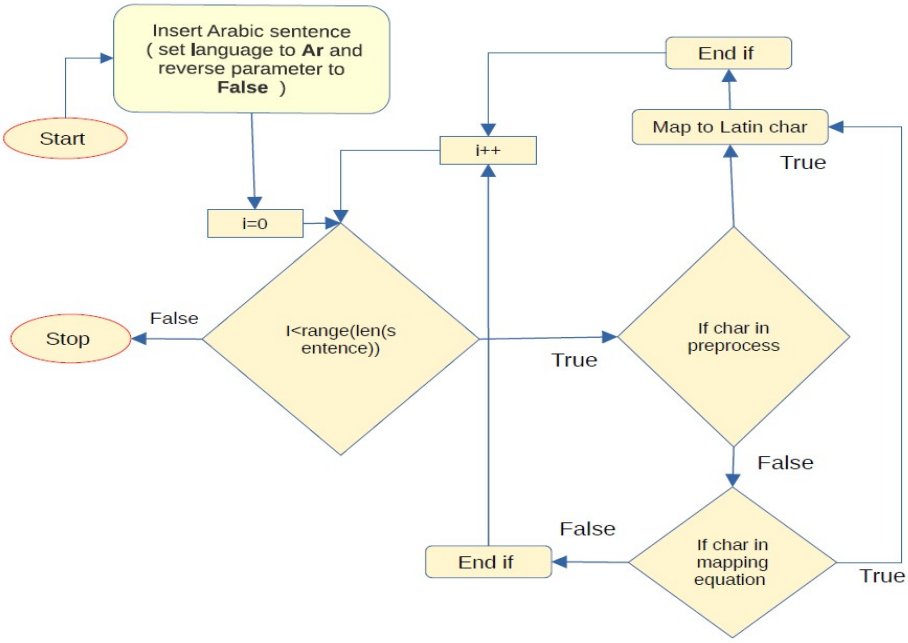

Fig. 1 Flow Chart for Conversion

3.6. *Converting franco-arabic to arabic*

In this subsection, the third step is considered. As mentioned above we can convert Arabic to Franco-Arabic and vice versa. But we just convert Arabic to Franco-Arabic and searched for another method to Convert Franco-Arabic to Arabic to ensure that another mapping was used. This method guarantees showing errors due to ambiguity in mapping characters from one method to another.

There are a few methods to translate Franco-Arabic to Arabic. An online startup called Yamil [24] introduced a free product called "the smart Arabic keyboard" that enables users to convert Arabic text written in

Latin letters into Arabic text written in Arabic characters. In other words, it enables users who are unfamiliar with the Arabic keyboard, to write in Arabic letters without using any Arabic letters.

A smart Arabic keyboard may be incorporated with any website using the Yamil Application Programmable Interface API to "Yamlify" all or some of the text boxes. In terms of the settings of Yamlified text boxes, such as enabling/disabling Yamil or even the fonts and colors used, the developer's preferences can be updated in the API's code. Unfortunately, there was a problem with using the Yamli API because it yamlifies text regions rather than strings. This indicates that the API was created to yamlify words, not complete sentences, as they are written by a user into a yamlified text field. In other words, the Yamil API relies on the user's keyboard and mouse event listeners, whereas when the user inputs and pushes "space," for instance, a single word is yamlified. However, the sentence is preserved if the user "paste" the entire thing rather than inputting it word for word.

To enable the translation of the entire statement, we wrote code that changes the behavior of Yamli open-source mechanism of operation. Due to those modifications in the source code, it loops over the entire text and generates objects for each word. These objects are then passed to the yamlification functions, where each word is yamlified separately. This amplifies the entire statement and sets each word's translation to the first alternative on the list before returning the entire sentence in Arabic characters.

3.7. *Catching the errors in converted sentences*

In this subsection, the last three steps are discussed. After converting Franco-Arabic to Arabic we need a tool to loop over the words in each sentence, detect error words and suggest corrections. We used PyEnchant [25] which is a spellchecking library for Python, based on the excellent Enchant [26] library. It enables us to use a dictionary to detect error words and recommend corrections. Because we are dealing with an Egyptian dialectal that is not MSA and this is an obstacle because most of the Dictionaries are built using MSA or classical Arabic. We used the Egyptian twitter dataset to extract the dictionary of Egyptian slang that helped us to detect error words. When we try to detect wrong/misspelled words we faced a problem is that PyEnchant flags names as misspelled words. We must exclude names like (persons – cities - countries - …) from the sentence before detecting error words.

There are many types of research in NLP branches that used BERT. At that point, we are interested in the named entity recognition NER topic [27] for the Arabic language. It is one of the most common data preparation tasks. It locates valuable information in the text and classifies it into several predetermined categories. We used the pre-trained CAMeLBERT-Mix NER Model [28] to detect names in our sentence and exclude it before identifying error words and introducing corrections. At that moment we know the error words in the sentence and have suggested corrections. The next step is to replace every wrong word with a [MASK] token. We will use pre-trained models for the Arabic language based on Transformers like MARBERT [29] and ARABERT [30]. It will be used to predict the original value of the masked words, based on the context provided by the other words, i.e., those non-masked words, in the sequence.

MARBERT model uses the same network architecture of ARABERT which is BERT-base but uses a different dataset consisting of both Dialectal Arabic (DA) and MSA. A random sample of 1 billion Arabic tweets from a sizeable in-house dataset of about 6 billion tweets was used to train the MARBERT model. We used the MARBERT model to predict masked words because MARBERT trained by Dialectal Arabic gives the best prediction side by side with PyEnchant's suggested correction. We compare two corrections and measure the similarity between them. If the similarity value is larger than 70% we used the correction of MARBERT else we will use PyEnchant's suggested correction. We manually reviewed and annotate the similarity score between the original Arabic text and the text resulting after applying previous steps we find that 455 records of the resulting texts are identical to the original text with an accuracy of **91%.** However, the previous steps give reasonable results but they have some drawbacks due to the number of steps that consume time and the difficulty of building dictionaries.

4. Building Model from Scratch

As mentioned above a new era started when google researchers introduce the BERT [20] model as a language model, when trained on a big corpus shown quite effective at interpreting English. For most NLP tasks, these models were able to innovate and produce state-of-the-art outcomes.

There are four variations of the basic BERT model, see Table 6 for detailed information [31]. The BERT model's Uncased and Cased versions denote whether the text was lowercase before tokenization and the text's case, respectively. The number of encoder layers separates BERT basic from BERT large. The BERT large model contains 24 layers of encoders layered on top of each other, compared to the BERT basic model's 12 layers. The BERT-based model uses the encoder-decoder transformer design.

Transformers [19] use feed-forward and skipping mechanisms to implement various layers of multi-head self-attention. The multi-head feature allows each layer to pay attention to distinct words within the input

sequence of text while the multi-head self-attention just pays attention to the input sequence of text. The order of the input sequence, the location of the word within the sequence, and the separation between words are all represented as vectors by the positional encoding technique and added to the embedding layer. These vectors help in capturing the contextual information within the input sequence. Following each self-attention layer is a residual connection, which is represented by a normalization layer. This normalization layer adds the input vector of the self-attention layer to the output vector from the same self-attention layer, assisting in the transfer of forgotten information to the following layers.

Table 6. The Versions of The Bert Model

BERT version	Layers	Hidden	Heads	Parameters
BERT-Large (Uncased or cased)	24	1024	16	340 M
BERT-Base, (Uncased or cased)	12	768	12	110 M

4.1. *Tokenization*

As mentioned above the best strategy to deal with Franco-Arabic is dealing with the original source directly because this method ensures that the sentiments and meanings of the original text are preserved. So we converted our collected dataset and applied pre-processing technique mentioned above then map Arabic To Franco-Arabic. Tokenization is a fundamental pre-processing step for the majority of NLP applications. It involves breaking text down into smaller parts called tokens in order to transform an unstructured input string into a sequence of discrete components appropriate for a Machine Learning (ML) model (for example, words or word segments). Each token is converted into an embedding vector and fed into models that use deep learning (like BERT).

Breaking text into words is a basic tokenization strategy. However, this method treats words that are not part of the vocabulary as "unknown". To solve this problem, modern NLP models tokenize text into smaller word units that frequently nevertheless have linguistic meaning e.g., morphemes. So, even if a word is unknown to the model, specific sub-word tokens may still include enough details to allow the model to infer the meaning somewhat. WordPiece[32] is a well-known sub-word tokenization algorithm that can be used with numerous other NLP models like BERT.

To tokenize a given text into WordPieces, WordPiece first pre-tokenizes the text into words by dividing punctuation and white spaces. The BERT model is designed in such a way that the sentence starts with the [CLS] token and ends with the [SEP] token. At the start of the first sentence, a [CLS] token is added to the input word tokens, and at the end of each sentence, a [SEP] token is added. Also, segment embeddings in which each token has a marker designating either Sentence A or Sentence B. As a result, the encoder can discriminate between different phrases. The next step is to add positional embeddings to each token to show where it belongs in the phrase. Fig.2 adopted from [20] to show Franco-Arabic input representation for BERT. The BERT tokenizer gives the best results with 50000 vocab size and min frequency equal to 2.

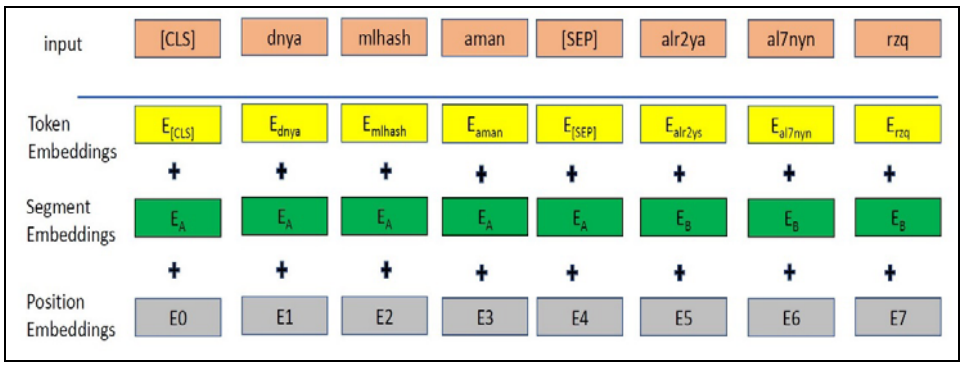

Fig. 2. BERT input representation for Franco-Arabic

4.2. *Training*

The original BERT is trained on two tasks: Predicting randomly masked tokens (MLM) and predicting whether two sentences follow each other next sentence prediction (NSP) [20]. The MLM task works by randomly masking off 15% of the input's words and replacing them with [MASK] tokens. The BERT attention-based encoder anticipates only the masked words using the context of the other non-masked words in the full sequence. During training, the BERT loss function only considers the prediction of the masked tokens and ignores the prediction of the non-masked tokens. As a result, the model converges significantly more gradually than right-to-left or left-to-right models. The BERT training procedure makes additional use of the next sentence

prediction to comprehend the relationship between two sentences. A pre-trained model with this kind of expertise is useful for doing activities like answering questions. The model learns to predict whether the second sentence is the next one in the original text when given pairs of sentences as input during training.

The BERT base architecture is used to train our model. We focused on the MLM task only because the sentences are short. After model training, we selected 7 sentences that include identical words used in different semantic meanings to analyze the ability of the model to understand Franco-Arabic. Table 7 shows sentences meaning in Arabic, English and Franco-Arabic. The visualizing of the embedding vectors in 3D, 2D, and similarity matrix results between sentences are shown in Fig.3, Fig.4 and Table 8 consequently. The visualization of embedding vectors shows that the newly trained model can understand the semantics meaning of Egyptian Franco-Arabic. This model is faster and more accurate because the sentiments and meanings of the original text are preserved.

Table 7. The sentences meaning in Arabic, English and Franco-Arabic

Arabic	English	Franco-Arabic
أريد أن أكل الخبز	I want to eat bread	ana 3awz akl 3esh
أريد أن أكل تفاح	I want to eat an apple	ana 3awz akl tfa7
أستطيع أن أكل التفاح	I can eat apples	ast6e3 akl tfa7
أستطيع أن أكل الخبز	I can eat breads	ast6e3 akl 3esh
نريد أن نأكل الخبز	We want to eat breads	a7na 3awzen akl 3esh
أريد أن نأكل التفاح	We want to eat the apples	a7na 3awzen akl tfa7
سوف أذاكر بإجتهاد	I will study hard	ana akl alktab akl

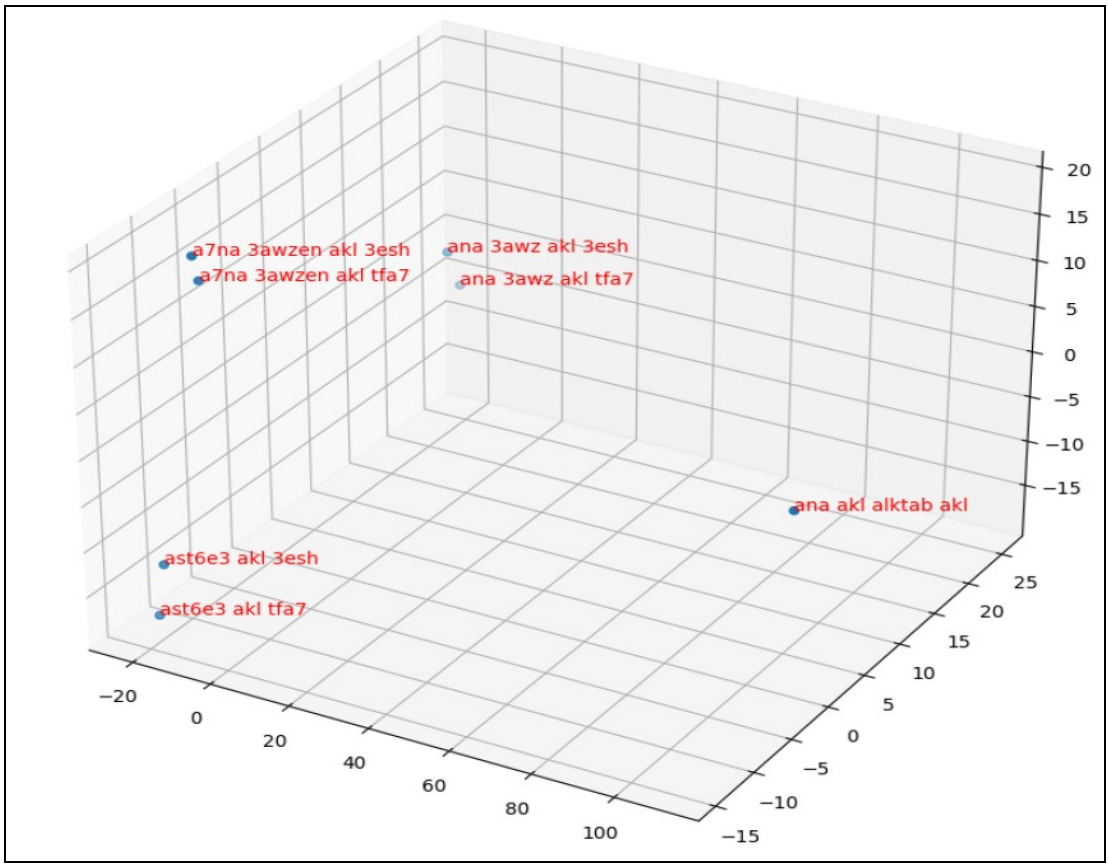

Fig. 4. 3D Vectors Visualization

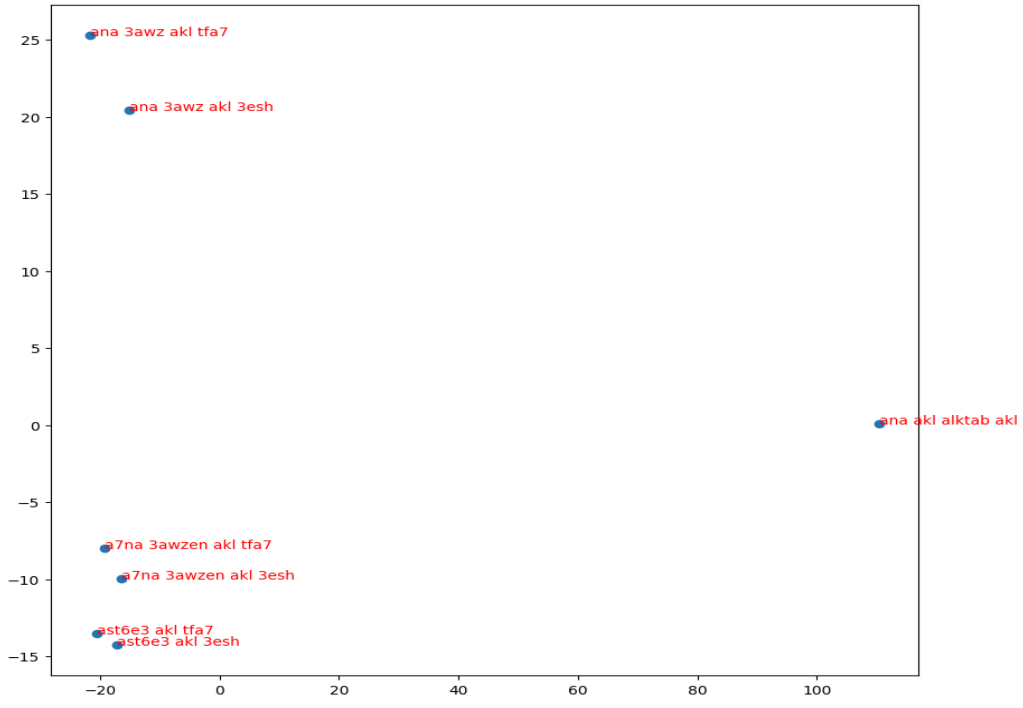

Fig. 5. 2D Vectors Visualization and Embedding Matrix

Table 8. The similarity matrix results between sentences

	ana 3awz akl 3esh	ana 3awz akl tfa7	ast6e3 akl 3esh	ast6e3 akl tfa7	a7na 3awzen akl 3esh	a7na 3awzen akl tfa7	ana akl alktab akl
ana 3awz akl 3esh	1.00	0.97	0.93	0.91	0.94	0.93	0.21
ana 3awz akl tfa7	0.97	1.00	0.91	0.92	0.90	0.93	0.12
ast6e3 akl 3esh	0.93	0.91	1.00	0.98	0.95	0.94	0.19
ast6e3 akl tfa7	0.91	0.92	0.98	1.00	0.93	0.95	0.15
a7na 3awzen akl 3esh	0.94	0.90	0.95	0.93	1.00	0.98	0.19
a7na 3awzen akl tfa7	0.93	0.93	0.94	0.95	0.98	1.00	0.17
ana akl alktab akl	0.21	0.12	0.19	0.15	0.19	0.17	1.00

5. Conclusion and Future Work

This research proposed two methods to process Egyptian Franco-Arabic texts. The First method reduces the errors that come due to ambiguity when converting Franco-Arabic texts to Arabic. The second method study deals with Egyptian Franco-Arabic directly using the trained model from scratch based on BERT base architecture. The results showed the ability of the new model to understand Franco-Arabic from the source. This

model can be fine-tuned for various NLP tasks. Of course, a larger dataset means enhancement of the model but requires more physical resources. Transformer-based models have successfully demonstrated their ability to deal with NLP problems and produce tangible results within a fleeting time. In the future work, we are targeting to convert massive Arabic data to Franco-Arabic and train transformers-based models on Arabic and Franco-Arabic with same model. This task will introduce to us a model able to deal with Franco-Arabic and Arabic texts using same model.

Conflicts of interest

The authors have no conflicts of interest to declare

References

[1] "Global Social Media Stats — DataReportal – Global Digital Insights." https://datareportal.com/social-media-users (accessed Feb. 28, 2022).
[2] "Top Ten Internet Languages in The World - Internet Statistics." https://www.internetworldstats.com/stats7.htm (accessed Dec. 02, 2022).
[3] M. Hijjawi, Z. Bandar, K. Crockett, and D. Mclean (2011), "An Arabic Stemming Approach using Machine Learning with Arabic Dialogue System," no. April, pp. 12–14, 2011.
[4] K. Darwish and W. Magdy(2014), "Arabic Information Retrieval Foundations and Trends R in Information Retrieval," Inf. Retr. Boston., vol. 7, no. 4, pp. 239–342, 2014.
[5] "Arabic chat alphabet - Wikipedia." https://en.wikipedia.org/wiki/Arabic_chat_alphabet (accessed Dec. 22, 2022).
[6] M. Magdy, C. Sabty, N. Sharaf, and S. Abdennadher(2018), "Mood extraction from Franco-Arabic Facebook Posts," no. March, 2018.
[7] "NLTK :: Natural Language Toolkit." https://www.nltk.org/ (accessed Nov. 27, 2022).
[8] T. H. A. Soliman, M. M. Ali, A. R. Hedar, and M. M. Doss, "MINING SOCIAL NETWOR KS ' ARABIC SLANG COMMENTS."
[9] M. C. Bousri, R. Bensalem, S. Bessa, Z. Lamri, C. Zakaria, and N. Bousbia (2023), Rumor Detection in Algerian Arabizi Based on Deep Learning and Associations, vol. 593 LNNS. Springer International Publishing, 2023.
[10] S. Hajbi, Y. Chihab, R. Ed-Dali, and R. Korchiyne(2022), Natural Language Processing Based Approach to Overcome Arabizi and Code Switching in Social Media Moroccan Dialect, vol. 357 LNNS. Springer International Publishing, 2022.
[11] C. Fourati, H. Haddad, A. Messaoudi, M. B. H. Hmida, A. B. E. Mabrouk, and M. Naski (2021), "Introducing A large Tunisian Arabizi Dialectal Dataset for Sentiment Analysis," WANLP 2021 - 6th Arab. Nat. Lang. Process. Work. Proc. Work., pp. 226–230, 2021.
[12] B. Talafha, A. Abuammar, and M. Al-Ayyoub(2021), "ATAR: Attention-based LSTM for Arabizi transliteration," Int. J. Electr. Comput. Eng., vol. 11, no. 3, pp. 2327–2334, 2021, doi: 10.11591/ijece.v11i3.pp2327-2334.
[13] H. P. Luhn, "A Statistical Approach to Mechanized Encoding and Searching of Literary Information*."
[14] W. A. Qader, M. M. Ameen, and B. I. Ahmed, "An Overview of Bag of Words;Importance, Implementation, Applications, and Challenges," Proc. 5th Int. Eng. Conf. IEC 2019, no. July, pp. 200–204, 2019, doi: 10.1109/IEC47844.2019.8950616.
[15] G. Kowalski(1997), Information retrieval systems : theory and implementation. Kluwer Academic Publishers, 1997.
[16] T. Mikolov, K. Chen, G. Corrado, and J. Dean(2013), "Efficient estimation of word representations in vector space," 1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc., pp. 1–12, 2013.
[17] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation." [Online]. Available: http://nlp.
[18] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov(2017), "Enriching Word Vectors with Subword Information," Trans. Assoc. Comput. Linguist., vol. 5, pp. 135–146, 2017, doi: 10.1162/tacl_a_00051.
[19] A. Vaswani et al.(2017), "Attention is all you need," Adv. Neural Inf. Process. Syst., vol. 2017-Decem, no. Nips, pp. 5999–6009, 2017.
[20] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova(2019), "BERT: Pre-training of deep bidirectional transformers for language understanding," NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf., vol. 1, no. Mlm, pp. 4171–4186.
[21] "Corpus on Arabic Egyptian tweets - Harvard Dataverse." https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/LBXV9O (accessed Nov. 27, 2022).
[22] "Twitter API Documentation | Docs | Twitter Developer Platform." https://developer.twitter.com/en/docs/twitter-api (accessed Mar. 05, 2023).
[23] "transliterate · PyPI." https://pypi.org/project/transliterate/ (accessed Mar. 11, 2022).
[24] "Yamli API." https://www.yamli.com/api/ (accessed Nov. 28, 2022).
[25] "PyEnchant — PyEnchant 3.2.2 documentation." http://pyenchant.github.io/pyenchant/ (accessed Nov. 28, 2022).
[26] "Enchant." https://abiword.github.io/enchant/ (accessed Nov. 28, 2022).
[27] "What is named entity recognition (NER) and how can I use it? | by Christopher Marshall | super.AI | Medium." https://medium.com/mysuperai/what-is-named-entity-recognition-ner-and-how-can-i-use-it-2b68cf6f545d (accessed Nov. 28, 2022).
[28] "CAMeL-Lab/bert-base-arabic-camelbert-mix-ner · Hugging Face." https://huggingface.co/CAMeL-Lab/bert-base-arabic-camelbert-mix ner (accessed Nov. 28, 2022).
[29] M. Abdul-Mageed, A. R. Elmadany, and E. M. B. Nagoudi(2021), "ARBERT & MARBERT: Deep bidirectional transformers for Arabic," ACL-IJCNLP 2021 - 59th Annu. Meet. Assoc. Comput. Linguist. 11th Int. Jt. Conf. Nat. Lang. Process. Proc. Conf., no. ii, pp. 7088–7105, 2021, doi: 10.18653/v1/2021.acl-long.551.
[30] W. Antoun, F. Baly, and H. Hajj(2020), "AraBERT: Transformer-based Model for Arabic Language Understanding," no. May, 2020, [Online]. Available: http://arxiv.org/abs/2003.00104.
[31] M. Hammad, M. Al-Smadi, Q. B. Baker, and S. A. Al-Zboon(2021), "Using deep learning models for learning semantic text similarity of Arabic questions," Int. J. Electr. Comput. Eng., vol. 11, no. 4, pp. 3519–3528, Aug. 2021, doi: 10.11591/ijece.v11i4.pp3519-3528.
[32] "A Fast WordPiece Tokenization System – Google AI Blog." https://ai.googleblog.com/2021/12/a-fast-wordpiece-tokenization-system.html (accessed Mar. 05, 2023).

Author's Profile

Mohamed Abdelnabi Ibrahim, he is currently pursuing a master's at the faculty of engineering at Ain shams university. Also, he obtained his B.C.A degree from a military-technical college in 2013 and he worked as a freelancer engineer in the fields of programming and artificial intelligence.

Ahmed Zaki Badr, holds a master's degree in computer science and engineering from Ain Shams University in 1982, and a Ph.D. in the same specialty in 1986. He also held the position of Minister of Education in Egypt and a Researcher at the National Institute of Engineering, Grenoble, France

Hani Mohamed Kamal Mahdi, He holds a master's degree in computer science and engineering from Ain Shams University in 1976 and a Ph.D. in Computer Science Engineering from TU Braunschweig: Institute of Communication Technology, Germany in 1984. He has published over sixty peer papers in reputed international journals and conferences. His current research interests include mathematics, artificial intelligence, and machine learning.