Evaluating Modeled Arabic-English Code Switching Using Quantitative Analysis With Python Word2Vec

John Speaks
jspeaks2@illinois.edu

Abstract

This paper evaluates the ability of GPT-4, a state-of-the-art large language model, to emulate Arabic-English code-switching, a complex sociolinguistic phenomenon prevalent in multilingual communities. Code-switching involves alternating between languages in a single discourse, influenced by social, contextual, and linguistic factors. The study compares synthetic code-switching generated by GPT-4 with naturally occurring examples, leveraging Word2Vec models trained using both CBOW and skip-gram techniques. Through quantitative analysis of word embedding similarities, the research identifies significant differences in language use patterns between synthetic and natural datasets. These differences suggest that current models, despite their advanced capabilities, do not fully replicate the nuanced utterances of real-world code-switching. This paper highlights the need for more localized and contextually informed datasets to enhance model accuracy. Future directions include refining datasets to better align linguistic registers and dialectal contexts for more precise evaluations.

1 Introduction

The aim of this paper is to explore English-Arabic code-switching and evaluate if state-of-the-art large language models (LLMs) like GPT-4 are capable of mimicking this sociolinguistic phenomenon. Code-switching, the practice of alternating between two or more languages within a single conversation or utterance, is a common feature in multilingual communities. For Arabic speakers, the interplay between Arabic and English is particularly significant given English's role as a global lingua franca and its widespread use in professional, academic, and online domains.

Arabic is spoken by over 400 million people worldwide and exists in a continuum of varieties, including Modern Standard Arabic (MSA), Egyptian Arabic, and numerous regional dialects. This linguistic diversity, combined with the social and functional roles of English, creates a rich environment for code-switching practices. The patterns and motivations behind these switches vary based on factors such as the speakers' geographical location, social context, and proficiency in each language.

Despite the richness of natural code-switching, synthetic data generated by models like GPT-4 may not fully capture the subtleties of such bilingual behavior. By comparing natural code-switching data with synthetically generated examples, this paper seeks to identify whether existing LLMs accurately represent the nuances of real-world language use. The study leverages quantitative methods, particularly word embedding models trained with Word2Vec, to analyze similarities and differences between natural and synthetic datasets.

Understanding the accuracy of LLM-generated code-switching is crucial for applications in natural language processing (NLP) such as conversational AI, educational tools, and cross-cultural communication platforms. Because English-Arabic code-switching is highly present online, new NLP tools should be able to understand and respond to this language use. Furthermore, this research contributes to the broader field of sociolinguistics by providing insights into how computational models encode and replicate complex linguistic phenomena.

2 Background on the Corpora

For this exploration, two corpora of English-Arabic code switching were used. One corpus was used to represent naturally occurring code-switching and the other is artificial code-switching generated by ChatGPT4.

The natural dataset comes from Hugging Face[1]. It is a mixture of several other datasets with differ-

[1] https://huggingface.co/datasets/MohamedRashad/arabic-english-code-switching

ent origins. The main segments of the data come from another project that was focused on identifying and manipulating Egyptian Arabic-English code-switching, so a large portion of this dataset is of that dialect. Another significant source of data is YouTube videos that contain Arabic-English code-switching. It was not specified what kinds of dialects are included in this data specifically, so for the purposes of this project it was assumed that they represent miscellaneous dialects of Arabic within the code-switching and thus the overall dataset is more heterogeneous. This dataset had approximately 12.5 thousand rows, where each row was one instance of a 1-3 sentence code-switched utterance.

The artificial dataset also comes from Hugging Face and the ArE-CSTD dataset[2]. This dataset was created by the National Center for Artificial Intelligence at the Saudi Data and Artificial Intelligence Authority (SDAIA) using the GPT-4 LLM. It contains a variety of test/train splits and distinct datasets including Egyptian dialects, Modern Standard Arabic (MSA) dialects, and Saudi Dialects. Due to the heterogeneous nature of the natural data, data from across the boundaries of the dataset were used to perform the evaluation. This artificial dataset was used in particular because it represents a state-of-the-art LLM that is in widespread use, and thus paints a good picture of the current use of this kind of language in the field. The subsequent evaluation then aims to analyze the generated code-switching produced by state-of-the-art models. This dataset was far larger than the natural one due to it being artificial, with 330 thousand rows each representing a sentence of code-switched text. There are a total of 3 million tokens in the data.

3 Preparing the Data

Prior to performing the analysis of this paper, it was necessary to clean and parse the datasets into clean and consistent forms. First, each dataset needed to be simplified to the same form before going through the main cleaning pipeline that was set up. The natural data also contained audio files for some samples, adding huge amounts of unneeded data to the file. This extraneous data was removed, and only the written text was preserved.

Next, both datasets went through a similar process. All punctuation marks were removed. This also included Arabic punctuation which had to be manually added to delimiters as they are not included by default in Python helper functions. After that, all strings were forced into being fully lower case. Finally, each sentence was split into individual word tokens by cutting at the space or new line characters. After this processing, the data had been parsed into a two dimensional array where for each word $x_{i,j}$ i represents the sentence of the original data, and j represents the word position.

There were some errors that would slip through this process anyway. For example, some English and Arabic words were not separated by a space because of how the invisible text direction changing characters function, and thus they would appear to be single token words. An attempt was made to remove these words, but because of the presence of some Arabizi and single word/expression code-switching in the data, this was removing real data as well. Because the amount of problematic data was very small and it did not affect downstream analysis, it was left within the data.

4 Methodology

To compare how code-switching is used in the two corpora, four Word2Vec embedding models were trained. A CBOW and skip-gram model were trained for each corpus to explore the differences in the corpora both through a more semantic lens with CBOW and a syntactic one with skip-gram. For all of the models, the data was first cleaned, as described in the previous section, to a consistent form. Next, the embedding spaces were trained with a 100-dimensional vector space and using a window size of 5 around each token. 50 epochs were used in training the embedding space to get a good model of the space, but at the same time to not overfit the data. This analysis ensures a robust understanding of how syntactic and semantic relationships vary not only within each dataset but also across datasets. By incorporating both syntactic (skip-gram) and semantic (CBOW) perspectives, the experiment aimed to capture the full spectrum of linguistic nuances in code-switching behavior. All of the trained models are available on the GitHub repository associated with this project[3].

After training the models, the trained spaces were studied in a number of ways. First, random words were sampled from the common words

[2] https://huggingface.co/datasets/SDAIANCAI/Ar-En-Code-Switching-Textual-Dataset

[3] https://github.com/JTSIV1/Comparing-Natural-and-Synthetic-Arabic-English-Codeswitching

across both datasets, and their 10 nearest neighbors (that were common to the vocabulary of both datasets) were listed. Using the 10 nearest neighbors for the same words across the 4 models gave a good peak into how the words were used differently in the data. Using these nearest neighbors, the Jaccard similarity between the four models were calculated. These are a good measure of how the models work relative to each other as it shows if models have similar distributions in similarity.

The next method of analysis was to sample 200 pairs of words from the common vocabulary of both datasets. Next, for each of the four models, the similarity of the pairings was saved, and then plotted. This measure was focused on natural versus synthetic data, not CBOW versus skip-gram, so the similarity was always plotted against the opposite type of data for the same model. For example, if similarity measures a, b, c, and d were taken for two words in the natural CBOW, natural skip-gram, synthetic CBOW, and synthetic skip-gram models respectively, (a,c) and (b,d) would be plotted as points. The idea with these created scatter plots was to discern if words were being used the same (and had similar similarities) across the datasets. For each scatter plot, a line of best fit, and a Pearson correlation coefficient was calculated to observe how the data behaves across the models.

All analysis code is also detailed in the accompanying GitHub repository for the paper[4].

5 Results and Evaluation

The first results to explore are the random word similarity scores and Jaccard similarity scores across the models. 20 random words were samples, but only a sample of that will be explored here. Figure shows the 10 nearest neighbors for each model for a sampled English word and a sampled Arabic word.

Both of these example words follow the same patterns as many other words in the dataset. The full data can be found in the project GitHub repository[5]. The natural models seem to have less mixing of languages, English words making up all of the similar words to ’feel’ in English, and Arabic words making up almost all of the common words to ’بالناس’. This is reflected in the original dataset. In the synthetic data, many more sentences switch back and fourth between English and Arabic every few words or each phrase, while in the natural dataset, the sentences generally have one switch and have longer strings of just one language. This is thus reflected in the data.

Nat_cbow	Nat_sg	Synth_cbow	Synth_sg
عجبتني	وحشة	مليئًا	مزدحمة
قراية	عمال	أرخص	welcoming
algorithms	ليها	مليئاً	lively
ليها	الخميس	cheaper	vendors
عالية	قراية	مزدحمة	above
ب	عجبتني	outstanding	والنّاس
تتعمل	صحابى	مزدحماً	landscapes
أحس	مميزات	mesmerizing	mesmerizing
أصحى	الشيخ	الذين	الزاهية
تايلاند	علاقة	مليئة	بالألوان

Figure 1: Similar words to ’بالناس’ across all four models

Nat_cbow	Nat_sg	Synth_cbow	Synth_sg
am	again	felt	relaxed
her	trying	am	energized
take	difficult	think	felt
got	responsible	أشعر	makes
when	happy	wonder	fulfilled
made	has	slept	energetic
might	might	feels	refreshed
know	child	are	like
never	does	feeling	نشيطاً
accept	being	بحس	accomplished

Figure 2: Similar words to ’feel’ across all four models

Looking at the model differences, the artificial model has the same kind of distribution that we would expect. The CBOW model has semantically similar words (including both Arabic and English synonyms) and the skip-gram model has words that would typically appear around it with a given word’s syntactic representation. The natural models do not necessarily look the same. First, neither of the natural models includes cross-language synonyms, and instead are relatively homogeneous lists. In addition, the words in English examples seem not to be as related to each other as the words in the Arabic examples. This is likely related to the much smaller bits of English that are included in the natural data when compared to the 1:1.25 ratio of English to Arabic tokens in the synthetic data.

Across the 20 samples, Jaccard similarity was measured for the models and the averages can be

[4] https://github.com/JTSIV1/Comparing-Natural-and-Synthetic-Arabic-English-Codeswitching

[5] https://github.com/JTSIV1/Comparing-Natural-and-Synthetic-Arabic-English-Codeswitching

seen below in figure 3.

Models	Avg. Jaccard Sim.
Nat_cbow & Synth_cbow	0.0161
Nat_sg & Synth_sg	0.0056
Nat_cbow & Nat_sg	0.1850
Synth_cbow & Synth_sg	0.1275

Figure 3: Average Jaccard similarity between sets of nearest neighbors for 20 random words

From the Jaccard similarity, it appears that the synthetic and natural data are not very similar at all. When looking at both CBOW models or both skip-gram models, the average similarity is about 15 times smaller that the average similarity for both synthetic or both natural models. This indicates that the underlying data is different and that the code-switching use is likely different.

To explore this further, the similarity scores for 200 word pairs was calculated for each model and plotted where the each axis represents the similarity that a particular pairing got relative to that axis model. Figure 4 shows the aggregated results for natural and synthetic irregardless of model.

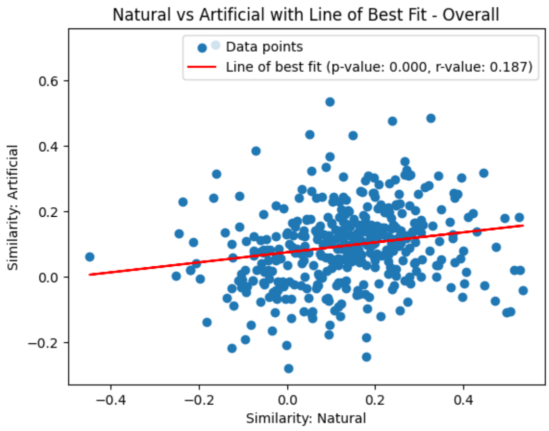

Figure 4: Plotted pairwise similarity for 200 random samples for aggregated natural and synthetic data

The data appears to be distributed fairly randomly. A line of best fit is plotted, but it have a very low r value of 0.187 indicating that the data is not very correlated. At the same time, the p-value is 0 indicating it is highly unlikely and virtually impossible that this distribution is random. To break this down further, figure 5 and 6 seperate the data points into CBOW and Skip-gram and then compare natural and synthetic data.

With these plots, the evidence for relationship between the similarity in the natural and synthetic data falls apart. Both of these plots have an r-value close to 0, indicating that there is little to no correlation. In addition, the p-value is higher than the typical cutoff of 0.05 indicating that there is not significant enough evidence to say that the similarities are correlated across natural and synthetic data for either model type and that the distribution of data could be due to random chance.

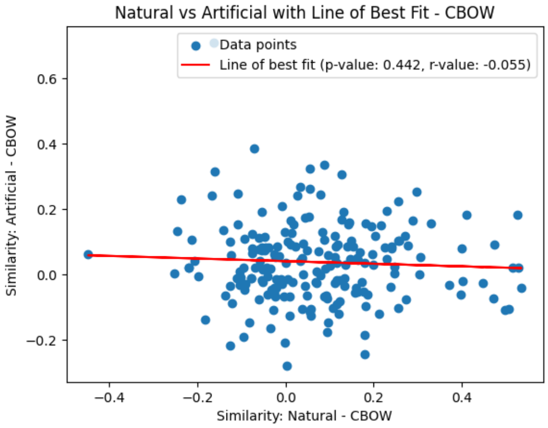

Figure 5: Plotted pairwise similarity for 200 random samples for CBOW models of natural and synthetic data

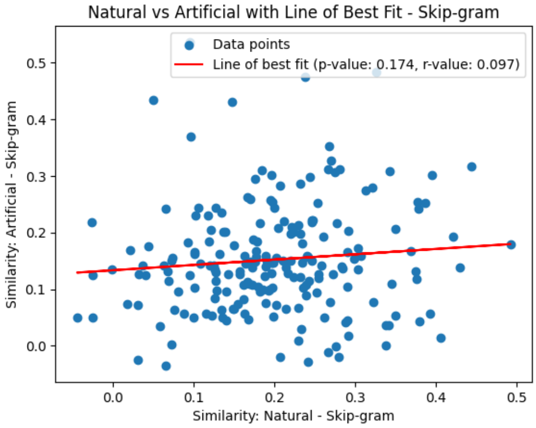

Figure 6: Plotted pairwise similarity for 200 random samples for skip-gram models of natural and synthetic data

This seems to indicate that the synthetically generated data is not following the same usage as organic, natural code-switching. To make sure that neither model is simply flawed and that there were no errors in creation, the correlation between natural CBOW and skip-gram was calculated and plotted as well as for the synthetic data. This verifies that the data is not correlated rather than the previous issues being due to a flawed model / data. The plots are below in figures 7 and 8.

Both of these figures have a middle-of-the-road

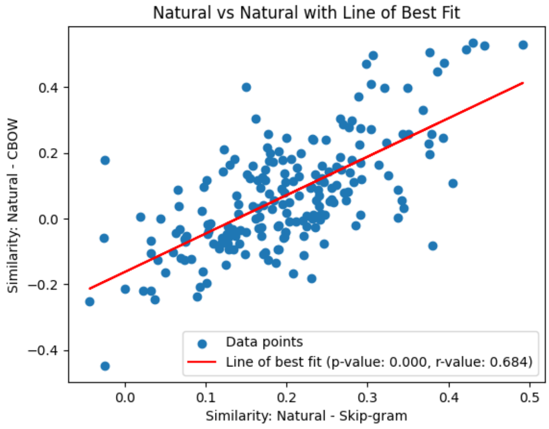

Figure 7: Plotted pairwise similarity for 200 random samples for CBOW and skip-gram models of natural data

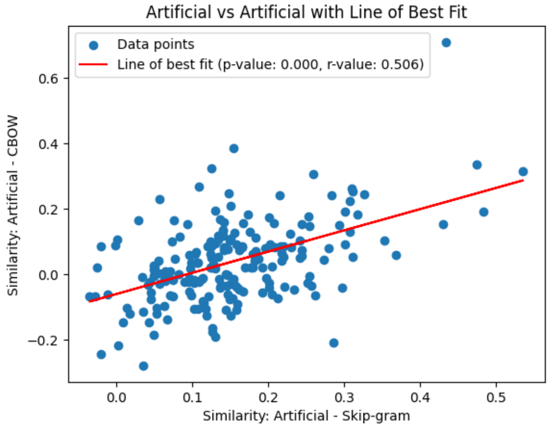

Figure 8: Plotted pairwise similarity for 200 random samples for CBOW and skip-gram models of natural data

r-value indicating some correlation, but more importantly have a p-value of near 0, indicating that the distribution is highly unlikely due to random chance and that the similarity is correlated somehow.

6 Limitations and Future Explorations

The main limitation in this paper originates from the natural dataset. While the synthetic dataset created using GPT-4 has over 330 thousand rows, and good metadata detailing the origin and varieties of language included, this is not true for the natural data. The natural data has much less data, and very little information about the origins and varieties of much of the data. Some of the data comes from YouTube, but it is unclear which dialect of Arabic is represented there. At the same time, the synthetic data was generated to mimic some register, but this register is not made clear by the publishers.

This is important because Arabic has a large amount of variation, especially in online registers. Code-switching has very localized practices, so even if the synthetic model is accurate to how some would naturally code-switch, it is possible it is code-switching in a different environment than the natural data, or the natural data contains unrelated samples.

Future studies should localize the code-switching data in order to compare it to synthetic data. This way, the lack of correlation could more conclusively say that GPT-4 is unable to mimic English-Arabic code-switching.

7 Conclusions

This study evaluated the ability of GPT-4 to generate realistic Arabic-English code-switching by comparing synthetic data to naturally occurring examples using Word2Vec embeddings. Our analysis revealed substantial differences between the natural and synthetic datasets, particularly in their linguistic structure and code-switching patterns. The Jaccard similarity scores and pairwise similarity correlations showed that synthetic data failed to replicate the nuanced usage of code-switching found in natural language.

The findings highlight a significant gap in current LLMs' ability to accurately model sociolinguistic phenomena like code-switching. While GPT-4 captures certain aspects of language mixing, it generates patterns that deviate from naturally occurring usage, potentially reflecting inconsistencies in its training data for Arabic-English code-switching or limitations in its architecture for this kind of sociolinguistic modeling. This suggests that while synthetic datasets may be valuable for certain applications, they should not be assumed to accurately reflect Arabic-English code-switching.

Future research should focus on improving the quality and diversity of training data used for synthetic language generation, especially in linguistically diverse communities. Additionally, localized studies that account for dialectal and contextual differences in code-switching practices are essential for a deeper understanding of the challenges and opportunities in this area. By addressing these limitations, NLP models could better serve multilingual users and applications, creating more inclusive and accurate language technologies.