# Performance Implications of Using Unrepresentative Corpora in Arabic Natural Language Processing

**Saied Alshahrani** **Norah Alshahrani** **Soumyabrata Dey** **Jeanna Matthews**
Department of Computer Science, Clarkson University, Potsdam, New York, USA
{saied, norah, sdey, jnm}@clarkson.edu

## Abstract

Wikipedia articles are a widely used source of training data for Natural Language Processing (NLP) research, particularly as corpora for low-resource languages like Arabic. However, it is essential to understand the extent to which these corpora reflect the representative contributions of native speakers, especially when many entries in a given language are directly translated from other languages or automatically generated through automated mechanisms. In this paper, we study the performance implications of using inorganic corpora that are not representative of native speakers and are generated through automated techniques such as bot generation or automated template-based translation. The case of the Arabic Wikipedia editions gives a unique case study of this since the Moroccan Arabic Wikipedia edition (ARY) is small but representative, the Egyptian Arabic Wikipedia edition (ARZ) is large but unrepresentative, and the Modern Standard Arabic Wikipedia edition (AR) is both large and more representative. We intrinsically evaluate the performance of two main NLP upstream tasks, namely word representation and language modeling, using word analogy evaluations and fill-mask evaluations using our two newly created datasets: Arab States Analogy Dataset (ASAD) and Masked Arab States Dataset (MASD). We demonstrate that for good NLP performance, we need both large and organic corpora; neither alone is sufficient. We show that producing large corpora through automated means can be a counter-productive, producing models that both perform worse and lack cultural richness and meaningful representation of the Arabic language and its native speakers.

## 1 Introduction

Natural Language Processing (NLP) plays a crucial role in decision-making systems. For instance, it is employed in resume parsers that assist in sorting job candidates. NLP systems are typically designed

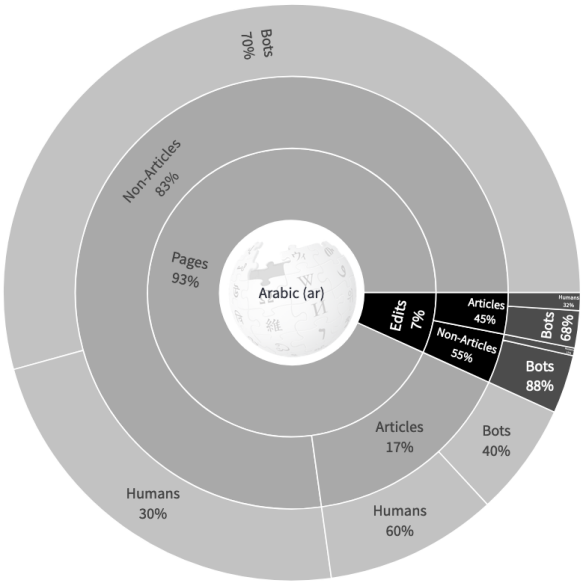



Figure 1: A sunburst visualization from our WIKIPEDIA CORPORA META REPORT dashboard (discussed in more detail in Appendix A) shows the percentage of contributions of bots and humans in the Modern Standard Arabic Wikipedia edition.

to analyze extensive collections of human text (corpora) with the goal of deriving insights from human behavior and generating recommendations on our behalf (Wali et al., 2020). The normal, organic, and representative corpora of human text produced by native speakers (the main ingredients in NLP systems) convey many social concepts, including culture, heritage, and even historic biases (Bolukbasi et al., 2016; Caliskan et al., 2017; Babaeianjelodar et al., 2020; Cho et al., 2021; Chen et al., 2021).

One of the widely used human text corpora and a common source of training data for NLP research is Wikipedia articles (content pages), especially in languages other than English. In specific, Wikipedia articles are used to train many Large Language Models (LLMs), such as ELMo (Embeddings from Language Models), which has been trained on the English Wikipedia and news crawl data (Peters et al., 2018); BERT (Bidirec-



*Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, pages 218–231

tional Encoder Representations from Transformers) has been trained on books with a crawl of English Wikipedia (Devlin et al., 2018); GPT-3 (Generative Pre-trained Transformer) has also been trained on five large datasets including the English Wikipedia (Brown et al., 2020); LaMDA (Language Model for Dialogue Applications) and PaLM (Pathways Language Model) were trained on a huge mixed dataset that includes English Wikipedia articles (Thoppilan et al., 2022; Chowdhery et al., 2022); and recently, LLaMA (Large Language Model Meta AI) was also pre-trained on the multilingual articles of Wikipedia from June to August 2022, covering 20 languages with a percentage of 4.5% of its overall training dataset size (Touvron et al., 2023).

Wikipedia corpora (editions) exist for over 300 of the over 7,000 languages spoken worldwide. These corpora vary greatly in size and quality, yet simply having a corpus of text in a certain language does not mean that it is an organic corpus representing the culture of native speakers. While native speakers originally write some corpora, others may be written by non-native speakers or translated from other languages (Nisioi et al., 2016). Recent research studied the Arabic Wikipedia editions: Modern Standard Arabic (AR), Egyptian Arabic (ARZ), and Moroccan Arabic (ARY), and found that in the Egyptian Arabic Wikipedia edition more than *one* million articles have been shallowly translated from English using either direct or template-based translation, all by a single registered user (Alshahrani et al., 2022). Alshahrani et al. (2022) argued that these shallowly translated articles do not echo the complex structure of the Arabic language and its dialects and do not express the views of Arabic speakers. In another recent research, Alshahrani et al. (2023) observed that the top ten Wikipedia editions (based on the total number of articles) are mostly bot-generated or auto-translated. To mitigate this problem, they introduced an enhanced Wikipedia depth metric, DEPTH$^+$, used as a rough indicator for the Wikipedia corpora quality, where they quantified and removed bot-generated Wikipedia articles and bot-made edits on those articles. Both works claimed that these practices of automation and translation could negatively impact the performance of NLP systems trained on these corpora, but they did not provide any empirical studies to show to which extent these practices could implicate the performance of specific NLP tasks and systems, including those using LLMs.

In this paper, we aim to bridge this gap by studying the performance implications of using such unrepresentative, inorganic corpora (produced by template-based translation or automatic bots creation/generation) by intrinsically evaluating two main NLP upstream tasks: word representation and language modeling, using word analogy and fill-mask evaluations, respectively, to capture syntactic and semantic relations between words. We purposely choose these intrinsic evaluations over extrinsic evaluations such as text classification or machine translation because many studies have shown that extrinsic and intrinsic evaluations' results are not consistently correlated, and the performance of NLP downstream tasks is always task-specific and can be significantly influenced by fine-tuning procedures (Faruqui et al., 2016; Schröder et al., 2021; Cao et al., 2022). We believe that evaluating NLP upstream tasks intrinsically will give us useful insights into the quality of the Arabic Wikipedia editions' corpora and show how the quality of corpora affects the performance of these NLP tasks.

We, in the following sections, discuss the problem of the unrepresentative corpora (§2), highlight the experimental setup of our study (§3), present the word representation and language modeling evaluations (§4 and §5), discuss the results and the limitations of our work (§6 and §7), provide a brief conclusion and offer future research ideas (§8).

## 2 Problem of Unrepresentative Corpora

The Wikipedia corpora (articles) unsurprisingly are not only used to train the large multilingual LLMs such as BERT (Devlin et al., 2018), LLaMA (Touvron et al., 2023), or even mGPT (multilingual GPT) (Shliazhko et al., 2022), but also have been used to train the majority of the Arabic LLMs, like AraBERT (Antoun et al., 2020), AraGPT2 (Antoun et al., 2021b), AraELECTRA (Antoun et al., 2021a), ARBERT and MARBERT (Abdul-Mageed et al., 2021), AraT5 (Nagoudi et al., 2022), *Jais* and *Jais-chat* (Sengupta et al., 2023), and recently, AceGPT (Huang et al., 2023). Therefore, there is a need to study Wikipedia's corpora representativeness, specifically in the Arabic Wikipedia editions, and to define the unrepresentativeness in its corpora as well. In this work, we generally define unrepresentative Wikipedia corpora as "*any Wikipedia articles (content pages) that have been created, generated, or edited without human involvement or supervision*", such as automatically created, gen-

erated, or edited Wikipedia articles using bots or shallowly template-translated articles from other highly resourced languages like English.

We study this problem from two perspectives: template-translated corpora and bot-generated corpora. For the template-translated corpora, Alshahrani et al. (2022) have studied the Arabic Wikipedia editions and shown that more than *one* million articles in the Egyptian Arabic Wikipedia have been directly translated using simple templates that lack rich content from the English language with the help of the off-the-shelf translation tools like Google Translate. These translation tools generally perform well, but not perfectly, and have several serious problems, such as gender bias, that could adversely affect the translated content (Prates et al., 2020; Ullmann and Saunders, 2021; Lopez-Medel, 2021). For the bot-generated corpora, a few recent research have shed light on the bots' activities on the Wikipedia project and their possible negative impacts on the quality of Wikipedia corpora (Tsvetkova et al., 2017; Zheng et al., 2019; Alshahrani et al., 2023). The root problem with the bots is that they can rapidly create Wikipedia articles (content pages) or edit the contents of those articles without any humans in the loop (Adler et al., 2008; Kang et al., 2021; Alshahrani et al., 2022).

| Wikipedia | Total Articles | Human created Articles (%) | Bot generated Articles (%) |
|---|---|---|---|
| **Arabic (AR)** | 1,197,467 | 717,678 (59.93%) | 479,789 (40.07%) |
| **Egyptian (ARZ)** | 1,616,530 | 1,616,515 (99.99%) | 15 (0.0001%) |
| **Moroccan (ARY)** | 6,426 | 5,684 (88.45%) | 742 (11.55%) |

Table 1: Categorization of Arabic Wikipedia editions by total articles, human-created articles, and bot-generated articles. This does *not* include the inorganic template-translated articles in the Egyptian Arabic Wikipedia.[1]

In this paper, we quantify the bots' activities in all Wikipedia editions and study the Arabic Wikipedia editions closely, specifically activities on their articles. We find that nearly 40% of articles in the Arabic Wikipedia edition are bot-generated (as demonstrated in Figure 1), and nearly 12% of articles in the Moroccan Arabic Wikipedia edition are bot-generated, as shown in Table 1. Surprisingly, the Egyptian Arabic Wikipedia edition has *only* 15 bot-generated articles, even though it is heavily affected by template-based translation activities (Alshahrani et al., 2022). We use Wikimedia `XTools` API[2] to identify Wikipedia articles' authors and exclude bot-generated articles from the Wikipedia corpora. We also use Wikipedia's "List Users" service[3] to retrieve the full list of bots in each Arabic Wikipedia edition to help us disclose the articles whose authors are in the bots list. We use the complete Wikipedia dumps of each Arabic Wikipedia edition, downloaded on the 1st of January 2023 (Wikimedia Foundation, 2023), process them using `Gensim` Python library (Řehůřek and Sojka, 2010), and preprocess them using `tr` Linux/Unix utility and `CAMeLTools` Python toolkit for Arabic NLP (Obeid et al., 2020). We extract all the Wikipedia articles from the three Arabic Wikipedia editions: Arabic, Egyptian Arabic, and Moroccan Arabic, and preprocess them slightly, removing the diacritical marks and the Latin letters and numbers; we do not apply stemming, lemmatization, or heavy text normalization on them to have organic texts (corpora) as much as possible.

### 2.1 Impact of Template-based Translation

Throughout this paper, to explore the impact of template-based translation, we compare the performance of models trained on the Egyptian Arabic Wikipedia edition's corpora that are dominated by shallow template-based translation (Baker, 2022; Alshahrani et al., 2022) to models trained on the Modern Standard Arabic and Moroccan Arabic Wikipedia editions' corpora, which are not.

### 2.2 Impact of Bot-based Generation

Similarly, throughout this paper, to explore the impact of bot-based generation, we compare the performance of models trained on Modern Standard Arabic and Moroccan Arabic Wikipedia editions' corpora (with and without bot-generated articles).

## 3 Experimental Setup

In this work, we examine two key NLP upstream tasks, namely word representation and language modeling, using curated corpora of the Arabic Wikipedia editions' articles and intrinsically evaluate them using two evaluation tasks on two newly created datasets. We next describe the evaluation tasks and our created datasets in more detail.

[1] Unlike the bots' quantifications process, the quantification of template-based translations is only specific to the Egyptian Arabic edition. Wikipedia project does not track template-based translation in its metadata as it does with bot generation.

[2] XTools API: https://www.mediawiki.org/wiki/XTools.

[3] https://{wiki_code}.wikipedia.org/wiki/Special:ListUser.

### 3.1 Evaluation Tasks

We use two evaluation tasks: word analogy and fill-mask, to intrinsically evaluate the two main NLP upstream tasks. In the following subsections, we describe these evaluation tasks in more detail.

#### 3.1.1 Word Analogy Task

The word analogy task was originally introduced by Mikolov et al. (2013a), and the goal is to find the missing word $b^\star$ in the relation: $a$ is to $a^*$ as $b$ is to $b^\star$, where $b$ and $b^\star$ are related by the same direction as $a$ and $a^*$. For example, king:*man*$^*$::queen:*woman*$^\star$. Each analogy question will be solved by calculating the target vector $b^\star$, $b^\star = b - a + a^*$. We calculate the cosine similarity between the target vector $b^\star$ and the vector representation of each word $w$ in a given word embedding vector $V$. We lastly get the most similar word $w$ to $b^\star$, following $\texttt{argmax}_{w \in V}(\texttt{sim}(w, b - a + a^*))$. If $w = b^\star$ (the same word), we then assume the given word embedding vector $V$ has answered the analogy question correctly.

We overcome the challenge of the Arabic words having possible multiple variants by 1) extending the top $K$ value (default $K$=1) to $K$={1, 5, 10} to search for the correct answer among the returned list of most similar words and 2) introducing a generic search algorithm that takes the word $w$ and then searches for all its possible variants. We only consider looking into the variants of *Alefs* {إ ، أ ، آ ، ا}, *Alef Maksura* {ى ، ي}, and *Teh Marbuta* {ه ، ة}. For example, if the word $w$ is "امرأة / woman", then the lookup list of $w$'s variants is: {إمرأة ، امرأه ، امراة ، امراه}.

#### 3.1.2 Fill-Mask Task

Masked language modeling involves masking some words in a sentence and predicting which words should replace those masked words. The valuable feature of this evaluation task is that it gives us a statistical understanding of the corpora on which our Masked Language Models (MLMs) are trained. We evaluate our MLM models that have been trained on the Arabic Wikipedia editions' corpora using our created datasets. We utilize the "fill-mask" pipeline of the Hugging Face with our MLM models (Wolf et al., 2020; Hugging Face, 2023a).

We follow the same approaches, as addressed in subsection 3.1.1, to beat the challenge of the Arabic words having possible multiple variants by extending the MLM top $K$ value (default $K$=10) to $K$={10, 50, 100} and using the previously introduced generic search algorithm that takes the word $w$ and searches for all its possible variants.

### 3.2 Created Datasets

We collect 20 Arab states with their corresponding capital cities, nationalities, currencies, and on which continents they are located.[4] We deliberately select the Arab states because they are facts and cannot change even in different Arabic dialects, like Egyptian and Moroccan Arabic. We, in the following subsections, describe these two created datasets in more detail.

#### 3.2.1 Arab States Analogy Dataset

We generate the Arab States Analogy Dataset (ASAD), consisting of four sets: country-capital set, country-currency set, country-nationality set, and country-continent set. Each set has 380 word analogies, and the total number of word analogies in the ASAD dataset is 1520. Table 2 demonstrates an example of each set, along with their English translations.

| ASAD Set | Word Analogy Example |
|---|---|
| **Country-Capital** | القاهرة مصر الرباط المغرب<br>Cairo Egypt Rabat Morocco |
| **Country-Currency** | مصر الجنيه المغرب الدرهم<br>Egypt Pound Morocco Dirham |
| **Country-Nationality** | مصر المصري المغرب المغربي<br>Egypt Egyptian Morocco Moroccan |
| **Country-Continent** | مصر أفريقيا المغرب أفريقيا<br>Egypt Africa Morocco Africa |

Table 2: Word analogy examples from the Arab States Analogy Dataset (ASAD) and their English translations.

#### 3.2.2 Masked Arab States Dataset

We generate the Masked Arab States Dataset (MASD), consisting of four categories: country-capital prompts, country-currency prompts, country-nationality prompts, and country-continent prompts. Each prompts category has 40 masked prompts, and the total number of masked prompts in the MASD dataset is 160. We notice that some masked prompts could lead to ambiguous masked prompts, which can be hard to be answered by the MLMs, and to fix this issue, we rephrase the ambiguous masked prompts, using

[4] We only drop two Arab states: the United Arab Emirates (الإمارات العربية المتحدة) and Comoros (جزر القمر), because they or their capital cities are written as open compound words (two words), like Abu Dhabi (أبو ظبي), which cannot be handled directly by the word embedding models.

the same facts/information about the Arab states. For example, the masked prompt "*The pound is the currency of <mask>.*" is ambiguous because many Arab states, including Egypt, Sudan, Lebanon, and Syria, use the pound as their currency, and our rephrase/disambiguation of this masked prompt is "*The currency of Egypt is the <mask>.*". Additionally, we add the masked prompts answers (masked words) of each masked prompt to the MASD dataset for the sake of validation and future evaluation. Table 3 shows an example of each masked prompts category, their masked prompts answers, and their English translations.

| MASD CATEGORY | MASKED PROMPTS EXAMPLE |
|---|---|
| **Country-Capital** | القاهرة هي عاصمة دولة <mask>.<br>Cairo is the capital of <mask>.<br>* **MASKED ANSWER:** مصر Egypt |
| **Country-Currency** | عملة دولة مصر هي <mask>.<br>The currency of Egypt is the <mask>.<br>* **MASKED ANSWER:** الجنيه Pound |
| **Country-Nationality** | أحب دولة مصر و أحب الشعب <mask>.<br>I love Egypt, and I love the <mask> people.<br>* **MASKED ANSWER:** المصري Egyptian |
| **Country-Continent** | تقع دولة مصر في قارة <mask>.<br>Egypt is located on the continent of <mask>.<br>* **MASKED ANSWER:** أفريقيا Africa |

Table 3: Masked prompts examples with their answers from the Masked Arab States Dataset (MASD) and their English translations.

## 4 Word Representation Evaluations

Word embeddings are a well-known word representation technique used by modern NLP systems as their backbone. They encode syntactic and semantic relations between words in a text and represent them in a low-dimensional space.

### 4.1 Impact of Template-based Translation

In the following subsections, we evaluate the performance of the word embedding models using the word analogy task and our ASAD dataset. Recall we compare the performance of models trained on the Egyptian Arabic Wikipedia edition's corpora, which are dominated by template-based translation, to the performance of models trained on Modern Standard Arabic and Moroccan Arabic Wikipedia editions' corpora, which are not.

#### 4.1.1 Word Embedding Models

We train *five* context-independent word embedding models on each Arabic Wikipedia edition's corpora using three different word representation algorithms: `Word2Vec` (continuous bag of words (cbow) and skip-gram), `fastText` (cbow and skip-gram), and `GloVe` (Mikolov et al., 2013b; Bojanowski et al., 2017; Pennington et al., 2014). We set these unified parameters of the three algorithms to these values: {*vector-size=300, epochs=20, window-size=2, min-count=1, alpha=0.03*}.

| WIKIPEDIA | ARTICLES | WORDS | SENTENCES |
|---|---|---|---|
| **AR** | 1,197,467 | 258,676,800 | 1,088,502 |
| **ARZ** | 1,616,530 | 65,565,053 | 728,340 |
| **ARY** | 6,426 | 720,334 | 5,394 |

Table 4: General statistics of the Arabic Wikipedia editions in terms of the total number of articles, total number of words, and total number of sentences.

Table 4 shows the Arabic Wikipedia editions' corpora statistics and confirms the findings of Alshahrani et al. (2022) that Egyptian Arabic Wikipedia has poor content pages, a side effect of the template-based translation. Although it has the largest number of articles among other Arabic Wikipedia editions, this large number of articles does not reflect the content richness when comparing the total words and sentences with the Modern Standard Arabic Wikipedia edition.

#### 4.1.2 Results of Word Analogy Task

We evaluate our word embedding models trained on the Arabic Wikipedia editions' corpora using our introduced ASAD dataset. In Table 5, we can see that increasing the top $K$ value and searching for words' variants improves the accuracy metric greatly. We also observe that the overall performance of the word embedding models varies, where the word embedding models trained on the Arabic Wikipedia edition's corpora performs dramatically better despite having fewer articles than the Egyptian Arabic Wikipedia edition's corpora, which comes in second in terms of performance; this contradicts the common assumption of "*the more articles a Wikipedia edition has, the better the quality of its corpus*". The word embedding models trained on the Moroccan Arabic Wikipedia edition's corpora performed the worst since they have been trained on very small corpora (less than 6,500 articles). This illustrates our key observation that we need both large and organic corpora for good NLP performance; neither alone is sufficient. We further highlight the best and worst word embedding models in Appendix B.

### 4.2 Impact of Bot-based Generation

We, in the following subsections, compare the performance of word embedding models that have

| Wikipedia | Model | K=1 | K=5 | K=10 |
|---|---|---|---|---|
| AR | Word2Vec-cbow | 53.88% | 74.47% | 79.67% |
| | Word2Vec-skipgram | 53.82% | 71.91% | 76.64% |
| | fastText-cbow | 21.97% | 34.67% | 44.47% |
| | fastText-skipgram | 39.67% | 57.17% | 65.79% |
| | GloVe | 36.58% | 50.53% | 54.14% |
| ARZ | Word2Vec-cbow | 13.88% | 26.97% | 33.09% |
| | Word2Vec-skipgram | 5.00% | 9.08% | 11.05% |
| | fastText-cbow | 10.13% | 20.86% | 28.09% |
| | fastText-skipgram | 11.64% | 18.22% | 22.37% |
| | GloVe | 0.53% | 3.29% | 5.20% |
| ARY | Word2Vec-cbow | 1.91% | 5.86% | 8.22% |
| | Word2Vec-skipgram | 2.11% | 4.01% | 5.92% |
| | fastText-cbow | 1.71% | 4.41% | 6.38% |
| | fastText-skipgram | 3.68% | 9.87% | 14.61% |
| | GloVe | 0.13% | 0.53% | 0.66% |

Table 5: Overall performance of each word embedding model of the Arabic Wikipedia editions evaluated on all the sets of our ASAD dataset.

been trained on Arabic and Moroccan Arabic corpora (with and without bot-generated articles) using the word analogy task and our ASAD dataset.

#### 4.2.1 Word Embedding Models

We train *five* context-independent word embedding models on both Arabic Wikipedia and Moroccan Arabic Wikipedia editions' corpora (after excluding bot-generated articles)[5] using three different word representation algorithms: `Word2Vec` (cbow and skip-gram), `fastText` (cbow and skip-gram), and `GloVe` (Mikolov et al., 2013b; Bojanowski et al., 2017; Pennington et al., 2014). We use the same values for the unified parameters for the three algorithms, as illustrated in subsection 4.1.1. In Table 6, we highlight the Arabic Wikipedia and Moroccan Arabic Wikipedia corpora statistics in terms of the number of articles, words, and sentences after all bot-generated articles are eliminated.

| Wikipedia | Articles | Words | Sentences |
|---|---|---|---|
| AR | 717,678 | 250,378,412 | 847,387 |
| ARY | 5,684 | 694,756 | 4,673 |

Table 6: General statistics of the Arabic Wikipedia and Moroccan Arabic Wikipedia editions regarding the number of articles, total words, and total sentences after removing the bot-generated articles.

#### 4.2.2 Results of Word Analogy Task

We evaluate our word embedding models that have been trained on the Arabic Wikipedia and Moroccan Arabic Wikipedia editions' corpora using our introduced ASAD dataset. As highlighted in 4.1.2, increasing the top $K$ value and searching for words' variants boosts the accuracy metric for the overall performance of all word embedding models of the Arabic and Moroccan Arabic Wikipedia editions. In Table 7, we compare the word embedding models trained on the Arabic Wikipedia corpora with bot activities (bot-generated articles included) and without bot activities (bot-generated articles excluded). We can see that most of the word embedding models trained with no bot-generated articles excel when $K$=1 and perform close to those trained with bot-generated articles when $K$={5, 10}. Surprisingly, the performance is generally the same or at times, even better, even though we have removed nearly 480K bot-generated articles (40% of total articles). This result emphasizes our observation that automated generation to increase the size of a corpus can actually be a counter-productive to NLP performance.

| AR Model | Corpora | K=1 | K=5 | K=10 |
|---|---|---|---|---|
| Word2Vec-cbow | With bots | 53.88% | 74.47% | 79.67% |
| | No bots | 53.22% | 74.47% | 79.47% |
| Word2Vec-skipgram | With bots | 53.82% | 71.91% | 76.64% |
| | No bots | 54.47% | 71.84% | 75.92% |
| fastText-cbow | With bots | 21.97% | 34.67% | 44.47% |
| | No bots | 22.76% | 34.34% | 43.29% |
| fastText-skipgram | With bots | 39.67% | 57.17% | 65.79% |
| | No bots | 39.87% | 56.64% | 67.43% |
| GloVe | With bots | 36.58% | 50.53% | 54.14% |
| | No bots | 38.29% | 52.11% | 55.13% |

Table 7: Overall performance of word embedding models of the Arabic Wikipedia edition evaluated on all the sets of our ASAD dataset before and after removing bot-generated articles.

| ARY Model | Corpora | K=1 | K=5 | K=10 |
|---|---|---|---|---|
| Word2Vec-cbow | With bots | 1.91% | 5.86% | 8.22% |
| | No bots | 1.84% | 4.54% | 7.11% |
| Word2Vec-skipgram | With bots | 2.11% | 4.01% | 5.92% |
| | No bots | 2.11% | 3.75% | 5.53% |
| fastText-cbow | With bots | 1.71% | 4.41% | 6.38% |
| | No bots | 1.97% | 4.41% | 6.45% |
| fastText-skipgram | With bots | 3.68% | 9.87% | 14.61% |
| | No bots | 3.62% | 9.54% | 13.75% |
| GloVe | With bots | 0.13% | 0.53% | 0.66% |
| | No bots | 0.07% | 0.26% | 0.39% |

Table 8: Overall performance of word embedding models of the Moroccan Arabic Wikipedia edition evaluated on all the sets of our ASAD dataset before and after removing bot-generated articles.

In Table 8, we also compare the performance of the word embedding models trained on the Moroccan Arabic Wikipedia corpora with bot activities (bot-generated articles included) and without bot activities (bot-generated articles excluded). We find that most of the word embedding models trained with bot-generated articles are generally better, except for the word embedding models produced by the `fastText` (cbow) that trained on no bot-generated articles (1.97% and 6.45% when $K$={1, 10}, respectively). We attribute these poor results to the small size of the Moroccan Arabic Wikipedia corpora, and eliminating the bot-generated articles makes the corpora even smaller. Once again, we say for good NLP performance, both large and organic corpora are very important.

[5] We drop the Egyptian Arabic Wikipedia due to having an insignificant number of bot-generated articles, only 15 articles.

## 5 Language Modeling Evaluations

Language modeling is an NLP task that generally predicts words in a sentence, and it is the heart of most existing LLMs. Some of these powerful LLMs, like BERT or RoBERTa, are usually trained using two objectives: masked language modeling and next sentence prediction (Devlin et al., 2018; Liu et al., 2019). In the following subsections, we exploit the masked language modeling objective in training Masked Language Models (MLMs) to produce contextual word embeddings and evaluate the performance of the MLM models trained on the Arabic Wikipedia editions' corpora using our created masked prompts dataset. We evaluate the quality of these MLM models using the Pseudo-Perplexity metric; we detailedly describe the evaluation process in Appendix C.

### 5.1 Impact of Template-based Translation

We, in the following subsections, evaluate the performance of the masked language models using the fill-mask task and our MASD dataset. Recall we compare the performance of models trained on the Egyptian Arabic Wikipedia edition's corpora, which are dominated by template-based translation, to the performance of models trained on Modern Standard Arabic and Moroccan Arabic Wikipedia editions' corpora, which are not.

#### 5.1.1 Masked Language Models

We train *three* RoBERTa$_{\text{BASE}}$ models *from scratch* on each Arabic Wikipedia edition's corpora (arRoBERTa$_{\text{BASE}}$, arzRoBERTa$_{\text{BASE}}$, and aryRoBERTa$_{\text{BASE}}$) with one modification on their architectures. We set the number of hidden layers to 6 instead of 12 for less computational overhead and to make the MLM models twice as fast as the RoBERTa$_{\text{BASE}}$ introduced by Liu et al. (2019).[6] We also train *three* Byte-level Byte-Pair-Encoding (BPE) tokenizers, one for each Arabic Wikipedia edition's corpora.[7] The full list of hyperparameters used to train our MLM models and tokenizers is shown in Table 9. We further evaluate these newly trained MLM models using the Pseudo-Perplexity metric in Appendix C.1.

[6]This modified architecture of RoBERTa$_{\text{BASE}}$ is called "DistilRoBERTa$_{\text{BASE}}$" by the Hugging Face: https://huggingface.co/distilroberta-base.

[7]We train our MLM models and their tokenizers using the Hugging Face Python libraries: `Transformers` and `Tokenizers` (Wolf et al., 2020). We exclude the default hyperparameters of training arguments from Table 9.

| RoBERTa$_{\text{BASE}}$ Model | Byte-level BPE Tokenizer |
|---|---|
| **Hidden Layers:** 6 | **Vocabulary Size:** 52,000 |
| **Hidden Size:** 768 | **Minimum Frequency:** 2 |
| **Attention Heads:** 12 | |
| **Vocabulary Size:** 52,000 | |
| **Type Vocabulary Size:** 1 | **Special Tokens:** |
| **Max Sequence Length:** 514 | • Start Token: <s> |
| **Number of Epochs:** 5 | • End Token: </s> |
| **Learning Rate:** 1e–4 | • Padding Token: <pad> |
| **Batch Size:** {128, 256} | • Unknown Token: <unk> |
| **Adam $\mathcal{E}$:** 1e–6 | • Masking Token: <mask> |
| **Adam $\beta_1$:** 0.9 | |
| **Adam $\beta_2$:** 0.98 | |
| **Weight Decay:** 0.01 | |
| **Trainable Parameters:** 83M | |

Table 9: Full list of hyperparameters of our Masked Language Models (MLMs) and their tokenizers.

#### 5.1.2 Results of Fill-Mask Task

We evaluate our MLM models that have been trained on the Arabic Wikipedia editions' corpora using our introduced MASD dataset. We can see in Table 10 that the performance of the Arabic arRoBERTa$_{\text{BASE}}$ model is superior to the Egyptian arzRoBERTa$_{\text{BASE}}$ model when $K$=10 (43.12% and 8.12%, respectively). Even though the Arabic Wikipedia edition has fewer articles than the Egyptian Arabic Wikipedia edition, it performs better and better represents the Arabic language. We also observe that increasing the MLM top $K$ value could lead to an average improvement in the performance of all MLM models, except the Moroccan aryRoBERTa$_{\text{BASE}}$ model, which scores zero accuracies regardless of the increment of the $K$ value; this is understandable since it was trained on corpora of less than 6,500 Wikipedia articles. Lastly, we see a performance jump of nearly 10% of the Egyptian arzRoBERTa$_{\text{BASE}}$ model when $K$={50, 100}, meaning the model is able to answer the masked prompts, but the correlation between the prompts and the answers is weak.

| MLM Model | K=10 | K=50 | K=100 |
|---|---|---|---|
| **arRoBERTa$_{\text{BASE}}$** | 43.12% | 45.00% | 50.62% |
| **arzRoBERTa$_{\text{BASE}}$** | 8.12% | 25.62% | 35.00% |
| **aryRoBERTa$_{\text{BASE}}$** | 0.00% | 0.00% | 0.62% |

Table 10: Performance of each masked language model of the Arabic Wikipedia editions on all the categories of MASD dataset.

### 5.2 Impact of Bot-based Generation

We, in the following subsections, compare the performance of masked language models that have been trained on Modern Standard Arabic Wikipedia and Moroccan Arabic Wikipedia editions' corpora (with and without bot-generated articles) using the fill-mask task and our MASD dataset.

#### 5.2.1 Masked Language Models

We train *two* RoBERTa$_{\text{BASE}}$ models *from scratch* on both Arabic Wikipedia and Moroccan Arabic Wikipedia editions' corpora after excluding bot-generated articles (arRoBERTa$_{\text{BASE}}$ and aryRoBERTa$_{\text{BASE}}$) and train *two* Byte-level Byte-Pair-Encoding (BPE) tokenizers, one for each Arabic Wikipedia edition's corpora; we drop the Egyptian Arabic Wikipedia for not having many bot-generated articles (only 15 articles). We use the same hyperparameters used to train our MLM models and tokenizers in subsection 5.1.1 and study the same processed corpora for Arabic Wikipedia and Moroccan Arabic Wikipedia, as discussed in Table 6, subsection 4.2.1. We further evaluate these newly trained MLM models using the Pseudo-Perplexity metric in Appendix C.2.

#### 5.2.2 Results of Fill-Mask Task

We evaluate our MLM models that have been trained on the Arabic Wikipedia and Moroccan Arabic Wikipedia editions' corpora (with and without bot-generated articles) using our introduced MASD dataset. As shown in Table 11, the MLM models trained on the Arabic Wikipedia corpora when bots' activities are eliminated (bot-generated articles) perform better than those trained on corpora that include the bots' activities, even though this corpus is smaller in terms of the number of articles than the corpora with bots. Interestingly, the performance of all Moroccan Arabic Wikipedia MLM models remains the same, even after being trained on no-bots corpora, which have fewer articles than the bots corpora.

| MLM Model | Corpora | K=10 | K=50 | K=100 |
|---|---|---|---|---|
| **arRoBERTa$_{\text{BASE}}$** | With bots | 43.12% | 45.00% | 50.62% |
| | No bots | 45.62% | 51.25% | 53.12% |
| **aryRoBERTa$_{\text{BASE}}$** | With bots | 0.00% | 0.00% | 0.62% |
| | No bots | 0.00% | 0.00% | 0.62% |

Table 11: Overall performance of MLMs of the Arabic Wikipedia and Moroccan Arabic Wikipedia editions evaluated on all the categories of MASD dataset before and after removing the bot-generated articles.

## 6 Discussion

Recent research has shown that not all Wikipedia editions (languages) are produced by native speakers, and there are substantial activities of auto-creation of articles (bot-generated articles) and auto-translation of articles (template-translated articles) in Wikipedia (Alshahrani et al., 2022, 2023). In this work, we argue that this automatic translation of articles, specifically the template-based translation on the Egyptian Arabic Wikipedia edition, impacts the overall performance of the NLP tasks due to having poor, limited, and unrepresentative corpora. Table 4 confirms that this template-based translation may enlarge the number of articles but cannot hide the true quality of a corpus. The Egyptian Arabic Wikipedia edition might have larger article numbers, but the truth is that these articles have fewer words and sentences than the Arabic Wikipedia edition. We find that all the word embedding models and all the masked language models that have been trained on each Arabic Wikipedia edition follow the same pattern, that is the models trained using the Arabic Wikipedia edition's corpora (which are widely believed to be mostly produced organically by the Arabic native speakers) perform better than the models trained on the Egyptian Arabic and Moroccan Arabic editions' corpora, as shown in Tables 5 and 10. We also believe that when $K$=10 (the default value), the masked language models usually show their actual performance, and as displayed in Table 10, it is obvious that the template-translated articles badly impact the masked language model trained on the Egyptian Arabic Wikipedia corpora when compared to the masked language model trained on the Arabic Wikipedia corpora despite the fact its corpora has nearly 480K articles more than the Arabic Wikipedia corpora, as shown in Table 4. It is evident that when masked language models are trained on naturally produced corpora by native speakers, they are more likely to have a better representation of the syntactic and semantic relations between words and a better understanding of the language itself and its native speakers.

We further argue, in this work, that the automatic creation and generation of articles, specifically the bots' creation and generation of articles on the Arabic Wikipedia and Moroccan Arabic Wikipedia editions, impacts the overall performance of the NLP tasks due to having unnatural, inorganic, and unrepresentative corpora. Once again, Table 1 confirms that this bots' generation may enlarge the number of total articles but cannot hide the true quality of a corpus. Even though the Arabic Wikipedia edition has a large number of articles (including bot-generated articles), the truth is that these bot-generated articles do not echo the complex structure of the Arabic language, do not reflect the cultural richness of the Arabic native speakers, and do not express the views of the Arabic native speakers.

We find that all the word embedding models that have been trained on the Arabic Wikipedia and Moroccan Arabic Wikipedia editions follow the same pattern, which is the models trained using the Arabic Wikipedia edition's corpora after eliminating the bot-generated articles, specifically when top $K$=1, perform better than the models trained on same corpora with bot-generated articles included, and of course, better than all models trained on the Moroccan Arabic Wikipedia edition's corpora, as shown in Tables 7 and 8. We believe when $K$=1 (the default value), the word embedding models usually show their actual performance, and as demonstrated in Table 7, it is obvious that the bot-generated articles negatively affect those word embedding models trained on them by widening the distance between words in the embedding space and that is why when we set $K$={5, 10}, those same word embedding models excel. We also find that all the masked language models trained on Arabic Wikipedia corpora perform better when all bot-generated articles are removed, indicating that, once again, the bots' creation or generation of articles negatively affects the masked language models, as demonstrated in Table 11.

Lastly, in this work, we strongly emphasize two points. First, we need both large and representative corpora to train NLP tasks and systems efficiently; neither alone is enough. The case of the Arabic Wikipedia editions gives a unique case study of this since the Moroccan Arabic Wikipedia edition is small but representative, and the Egyptian Arabic Wikipedia edition is large but unrepresentative. Second, removing many bot-generated articles from the Arabic Wikipedia corpora, for example, results in the same or even better performance. Due to the rise of generative models and for effective and safe training of NLP tasks and systems, we recommend avoiding using translated or generated corpora, especially when the goal is representation-based tasks like capturing the opinions or identifying the stances of Arabic native speakers.

## 7 Limitations

One limitation of our work is that while the three Arabic Wikipedia editions provide a unique example of our points, we cannot generalize the study and the impact of inorganic corpora for all the Wikipedia editions due to the lack of computational power needed to train the word embedding models and masked language models and due to the impracticality of creating or collecting factual datasets for the more than 300 languages that exist today on the Wikipedia project without using translation. Unlike the bots' quantifications process, the other limitation of our work is that the quantification of template-based translations is only specific to the Egyptian Arabic edition since the Wikipedia project does not track template-based translation in its metadata as it does with bot generation.

## 8 Conclusion and Future Work

In this work, we demonstrate that for good NLP performance, we need both large and organic corpora; neither alone is sufficient. We show that producing large corpora through automated means can be a counter-productive, producing models that both perform worse and lack cultural richness and meaningful representation of the Arabic language and its native speakers. Specifically, we demonstrate that training two key NLP upstream tasks, namely word representation and language modeling, on inorganic and unrepresentative corpora negatively impacts the performance of these NLP tasks. We find that the performance of these two NLP tasks is notably influenced by the way the training corpora are produced, where we observe that all models that have been trained on the template-translated corpora of the Egyptian Arabic edition perform the worst when compared with the more representative corpora like the Arabic Wikipedia edition. We also observe that many models perform the same or better when bot-generated articles are removed. Specifically, models trained on the Arabic Wikipedia edition (40% bot-generated articles) and Moroccan Arabic Wikipedia edition (12% bot-generated articles) perform the same or better when the bot-generated content is removed. In future work, we plan to expand our study of using unrepresentative corpora to include the societal implications (like gender bias and false representations) and security implications (like susceptibility to adversarial robustness) and hope to build a multi-level classification system to detect template-based translation activities such as those seen in the Egyptian Arabic Wikipedia edition.

## Reproducibility

We share our code scripts, created datasets, extracted corpora, and trained models on GitHub at https://github.com/SaiedAlshahrani/performance-implications.

## References


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

B. Thomas Adler, Luca de Alfaro, Ian Pye, and Vishwanath Raman. 2008. Measuring Author Contributions to the Wikipedia. In *Proceedings of the 4th International Symposium on Wikis*, WikiSym '08, New York, NY, USA. Association for Computing Machinery.

Saied Alshahrani, Norah Alshahrani, and Jeanna Matthews. 2023. DEPTH+: An Enhanced Depth Metric for Wikipedia Corpora Quality. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 175–189, Toronto, Canada. Association for Computational Linguistics.

Saied Alshahrani, Esma Wali, and Jeanna Matthews. 2022. Learning from Arabic corpora but not always from Arabic speakers: A case study of the Arabic Wikipedia editions. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 361–371, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based Model for Arabic Language Understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021a. AraELECTRA: Pre-Training Text Discriminators for Arabic Language Understanding. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 191–195, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021b. AraGPT2: Pre-Trained Transformer for Arabic Language Generation. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 196–207, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Marzieh Babaeianjelodar, Stephen Lorenz, Josh Gordon, Jeanna Matthews, and Evan Freitag. 2020. Quantifying Gender Bias in Different Corpora. In *Companion Proceedings of the Web Conference 2020*, WWW '20, page 752–759, New York, NY, USA. Association for Computing Machinery.

Maher Asaad Baker. 2022. *How I Wrote a Million Wikipedia Articles*, 2 edition. BookRix GmbH Co. KG., Munich, Germany.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Yang Trista Cao, Yada Pruksachatkun, Kai-Wei Chang, Rahul Gupta, Varun Kumar, Jwala Dhamala, and Aram Galstyan. 2022. On the Intrinsic and Extrinsic Fairness Evaluation Metrics for Contextualized Language Representations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 561–570, Dublin, Ireland. Association for Computational Linguistics.

Yan Chen, Christopher Mahoney, Isabella Grasso, Esma Wali, Abigail Matthews, Thomas Middleton, Mariama Njie, and Jeanna Matthews. 2021. Gender Bias and Under-Representation in Natural Language Processing Across Human Languages. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '21, page 24–34, New York, NY, USA. Association for Computing Machinery.

Won Ik Cho, Jiwon Kim, Jaeyeong Yang, and Nam Soo Kim. 2021. Towards Cross-Lingual Generalization of Translation Gender Bias. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 449–457, NYC, NY, USA. Association for Computing Machinery.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob

Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling Language Modeling with Pathways.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems With Evaluation of Word Embeddings Using Word Similarity Tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 30–35, Berlin, Germany. Association for Computational Linguistics.

Huang Huang, Fei Yu, Jianqing Zhu, Xuening Sun, Hao Cheng, Dingjie Song, Zhihong Chen, Abdulmohsen Alharthi, Bang An, Ziche Liu, Zhiyi Zhang, Junying Chen, Jianquan Li, Benyou Wang, Lian Zhang, Ruoyu Sun, Xiang Wan, Haizhou Li, and Jinchao Xu. 2023. AceGPT, Localizing Large Language Models in Arabic.

Hugging Face. 2023a. Fill-Mask. Last accessed on 2023-09-01.

Hugging Face. 2023b. Perplexity of fixed-length models. Last accessed on 2023-09-01.

Seonjun Kang, Xiaojin (Jim) Liu, Yeongin Kim, and Victoria Yoon. 2021. Can bots help create knowledge? The effects of bot intervention in open collaboration. *Decision Support Systems*, 148:113601.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Maria Lopez-Medel. 2021. Gender bias in machine translation: an analysis of Google Translate in English and Spanish. *Academia.edu*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient Estimation of Word Representations in Vector Space. *arXiv arXiv:1301.3781v3*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2022. AraT5: Text-to-Text Transformers for Arabic Language Generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 628–647, Dublin, Ireland. Association for Computational Linguistics.

Sergiu Nisioi, Ella Rabinovich, Liviu P. Dinu, and Shuly Wintner. 2016. A Corpus of Native, Non-native and Translated Texts. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, pages 4197–4201, Portorož, Slovenia. European Language Resources Association.

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. CAMeL Tools: An Open Source Python Toolkit for Arabic Natural Language Processing. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France. European Language Resources Association.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Marcelo OR Prates, Pedro H Avelar, and Luís C Lamb. 2020. Assessing gender bias in machine translation: a case study with Google Translate. *Neural Computing and Applications*, 32:6363–6381.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked Language Model Scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Sarah Schröder, Alexander Schulz, Philip Kenneweg, Robert Feldhans, Fabian Hinder, and Barbara Hammer. 2021. Evaluating Metrics for Bias in Word Embeddings. *arXiv arXiv:2111.07864v1*.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, Lalit Pradhan, Zain Muhammad Mujahid, Massa Baali, Alham Fikri Aji, Zhengzhong Liu, Andy Hock, Andrew Feldman, Jonathan Lee, Andrew Jackson, Preslav Nakov, Timothy Baldwin,

and Eric Xing. 2023. Jais and Jais-chat: Arabic-Centric Foundation and Instruction-Tuned Open Generative Large Language Models. *Inception, United Arab Emirates*.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2022. mGPT: Few-Shot Learners Go Multilingual. *arXiv preprint arXiv:2204.07580*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed H. Chi, and Quoc Le. 2022. LaMDA: Language Models for Dialog Applications. *CoRR*, abs/2201.08239.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models.

Milena Tsvetkova, Ruth García-Gavilanes, Luciano Floridi, and Taha Yasseri. 2017. Even good bots fight: The case of Wikipedia. *PloS one*, 12(2):e0171774.

Stefanie Ullmann and Danielle Saunders. 2021. Google Translate is sexist. What it needs is a little gender-sensitivity training. Last accessed on 2023-09-01.

Esma Wali, Yan Chen, Christopher Mahoney, Thomas Middleton, Marzieh Babaeianjelodar, Mariama Njie, and Jeanna Neefe Matthews. 2020. Is Machine Learning Speaking my Language? A Critical Look at the NLP-Pipeline Across 8 Human Languages. *arXiv preprint arXiv:2007.05872*.

Wikimedia Foundation. 2023. Wikimedia Downloads. Last accessed on 2023-09-01.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Lei (Nico) Zheng, Christopher M. Albano, Neev M. Vora, Feng Mai, and Jeffrey V. Nickerson. 2019. The Roles Bots Play in Wikipedia. *Proceedings of the ACM on Human-Computer Interaction*.

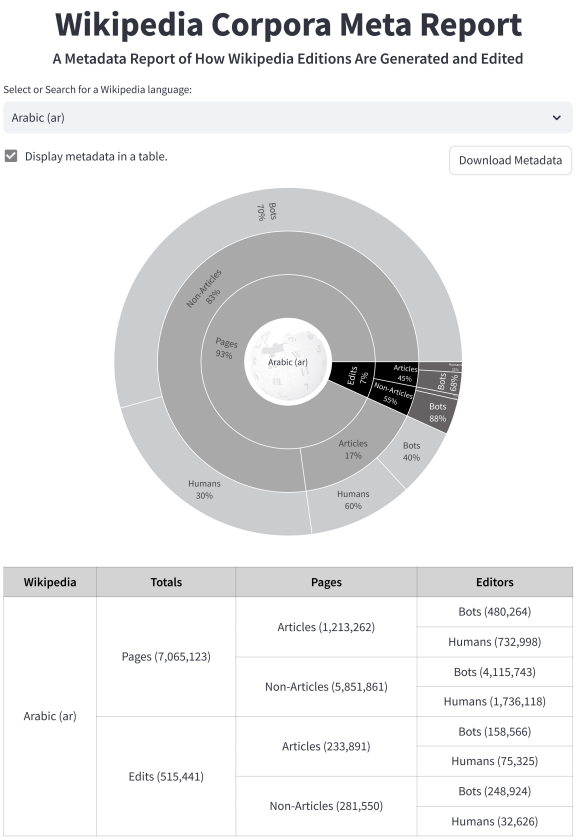
Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of LREC 2010 workshop New Challenges for NLP Frameworks*, pages 46–50, Valletta, Malta. University of Malta.

## A Wikipedia Corpora Meta Report

We release the WIKIPEDIA CORPORA META REPORT as an online metadata report (dashboard), designed to shed light on how bots or humans generate or edit Wikipedia editions to provide the NLP community with detailed information (metadata) about each Wikipedia edition's articles, enabling them to make informed decisions regarding using these Wikipedia articles for training their NLP tasks and systems. As demonstrated in Figure 2, the dashboard interactively displays the metadata of each Wikipedia edition using sunburst visualization and provides users with the options to view the metadata in a tabular format and to download the displayed metadata as a CSV file. The dashboard is open-sourced on GitHub with an MIT license at https://github.com/SaiedAlshahrani/Wikipedia-Corpora-Report and publicly hosted on Streamlit Community Cloud at https://wikipedia-corpora-report.app. In the following subsections, we briefly describe the system of the dashboard, outline its architecture, and discuss its limitations.

### A.1 System Description

The online WIKIPEDIA CORPORA META REPORT dashboard illustrates how humans and bots generate or edit Wikipedia editions, and calculates "pages" and "edits" metrics for all Wikipedia editions. The "pages" metric counts articles and non-articles, while the "edits" metric tallies edits on articles and non-articles, all categorized by contributor type: humans or bots. The dashboard dynamically displays these statistics using a sunburst visualization with three levels: metrics (pages or edits), sub-metrics (articles or non-articles), and contributors (bots or humans), showing numeric values and parent relationships at each level. Plus, the dashboard offers options to display metadata in a table format and allows users to download the metadata in CSV file format for their chosen Wikipedia edition/language.




| Wikipedia | Totals | Pages | Editors |
|---|---|---|---|
| Arabic (ar) | Pages (7,065,123) | Articles (1,213,262) | Bots (480,264) |
| | | | Humans (732,998) |
| | | Non-Articles (5,851,861) | Bots (4,115,743) |
| | | | Humans (1,736,118) |
| | Edits (515,441) | Articles (233,891) | Bots (158,566) |
| | | | Humans (75,325) |
| | | Non-Articles (281,550) | Bots (248,924) |
| | | | Humans (32,626) |

Figure 2: A screenshot of the online WIKIPEDIA CORPORA META REPORT dashboard, displaying a metadata report of how Modern Standard Arabic Wikipedia edition (AR) articles are generated and edited.

## A.2 System Architecture

The WIKIPEDIA CORPORA META REPORT dashboard comprises both front–end and back–end components, each with distinct functionality. Figure 3 illustrates the dashboard's architecture and workflow, emphasizing each component and its role.

### A.2.1 Front–end Components

The front–end components of this dashboard serve two specific functions: hosting the dashboard online for free public access and storing the metadata as a permanent Hugging Face dataset.

#### A.2.1.1 Streamlit Framework

We utilize the Streamlit Framework[8] to design, host, and deploy the dashboard on the free Streamlit Community Cloud[9] service, making it publicly accessible to everyone at https://wikipedia-corpora-report.streamlit.app.

[8]Streamlit Framework: https://streamlit.io.
[9]Streamlit Community Cloud: https://streamlit.io/cloud.

#### A.2.1.2 Hugging Face Datasets

We use Hugging Face Datasets[10] as our database to store the processed metadata. Simultaneously, the dashboard retrieves the metadata dataset from the Hugging Face Hub. The metadata dataset is available at https://huggingface.co/SaiedAlshahrani/Wikipedia-Corpora-Report.

### A.2.2 Back–end Components

The back–end components of this dashboard serve two specific functions: automatically updating the metadata dataset and triggering the metadata update procedure every 45 days.

#### A.2.2.1 Selenium WebDriver

We utilize the Selenium WebDriver[11] to automate the download of unprocessed metadata from the Wikimedia Statistics[12] service as CSV files. Then, we process the metadata and upload the processed metadata to the Hugging Face Hub as a dataset.

#### A.2.2.2 Unix/Linux Bash Daemons

We take advantage of the Streamlit Community Cloud being built on Debian Linux. We have written a Bash daemon that runs in the background and initiates the metadata update procedures. The daemon compares the original retrieval date from the pulled dataset with the system's current date, and when the time difference between these two dates exceeds 45 days, it triggers the update scripts.

## A.3 System Limitations

The limitation of the WIKIPEDIA CORPORA META REPORT is that we use the Wikimedia Statistics service to quantify the contributions of bots and humans to a specific Wikipedia edition. Yet, these quantifications are calculated statistically, meaning users cannot determine which Wikipedia articles have been generated or edited by bots or humans.

# B Best/Worst Word Embedding Models

We report that the `Word2Vec` (cbow) algorithm achieves the best accuracy when trained on substantially large corpora, like the Arabic and the Egyptian Arabic Wikipedia corpora (average accuracy: 69% and 25%, respectively), yet it does not when the corpora are very small, like the Moroccan Arabic Wikipedia corpora (average accuracy: 5%).

[10]Hugging Face Datasets: https://huggingface.co/datasets.
[11]Selenium WebDriver: https://selenium.dev/webdriver.
[12]Wikimedia Statistics service: https://stats.wikimedia.org.

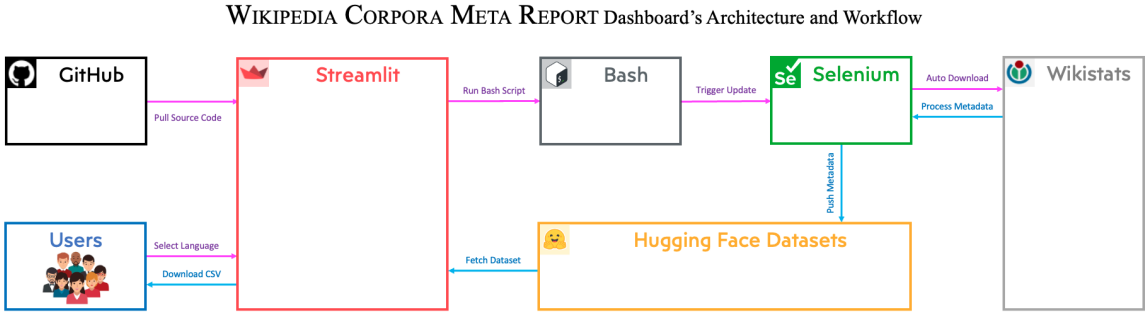



Figure 3: A diagram shows the WIKIPEDIA CORPORA META REPORT dashboard's architecture and workflow.

We also report that the GloVe algorithm achieves the lowest accuracy when trained on the Egyptian Arabic and the Moroccan Arabic Wikipedia corpora (average accuracy: 3% and 0.44%, respectively), yet it does the opposite when trained on corpora with lengthy articles, like the Arabic Wikipedia corpora (average accuracy: 47%).

## C Pseudo-Perplexity Evaluations

Perplexity (PPL) is a commonly used metric to evaluate the performance of language models, yet this PPL metric is mostly suitable for the classic/causal language models that predict the next word in a sentence and not a well-defined metric for the masked language models (Hugging Face, 2023b). Therefore, we evaluate our MLM models using the well-designed metric for the MLMs, the Pseudo-Perplexity (PPPL) metric, which is proposed by Salazar et al. (2020), to intrinsically measure how well MLMs model a corpus of sentences. We find that the calculations of the PPPL are susceptible to the length of the sentences, and to ensure accurate measurements, we randomly choose 500 sentences with character lengths between 400 and 500 from each Arabic Wikipedia edition.

### C.1 Impact of Template-based Translation

We calculate the PPPL scores for each MLM model, and in Table 12, we show the PPPL scores. We can see that the Arabic MLM (arRoBERTa$_{\text{BASE}}$) model, which has been trained on the Arabic Wikipedia edition, scores the best (the lower the PPPL score, the better the MLM model) with a PPPL score of 23.70, then the Egyptian Arabic MLM (arzRoBERTa$_{\text{BASE}}$) model with a PPPL score of 115.80, and lastly, the Moroccan Arabic MLM (aryRoBERTa$_{\text{BASE}}$) model with a very large PPPL score of 5,379.89. We attribute the high PPPL score of the aryRoBERTa$_{\text{BASE}}$ model to its very small training corpora (less than 6,500 articles) compared to the Arabic and Egyptian Arabic corpora. Still, we can also see a significant difference between the Arabic and the Egyptian Arabic MLMs' PPPL scores, indicating that even with a great number of articles, the documented template-based translation activity in the Egyptian Arabic Wikipedia edition seems to affect the performance of its MLM model.

| MLM MODEL | SAMPLES | PSEUDO-PERPLEXITY |
|---|---|---|
| **arRoBERTa$_{\text{BASE}}$** | 500 | 23.70 |
| **arzRoBERTa$_{\text{BASE}}$** | 500 | 115.80 |
| **aryRoBERTa$_{\text{BASE}}$** | 500 | 5,379.89 |

Table 12: Pseudo-Perplexity scores of all the Arabic Wikipedia editions' MLM models.

### C.2 Impact of Bot-based Generation

We evaluate our two MLM models (arRoBERTa$_{\text{BASE}}$ and aryRoBERTa$_{\text{BASE}}$) that have been trained on Arabic Wikipedia and Moroccan Arabic Wikipedia editions' corpora after excluding bot-generated articles using the PPPL metric. Table 13 displays that the PPPL measurements for the Arabic MML model (arRoBERTa$_{\text{BASE}}$) when trained once on corpora include bots activities, and trained another on corpora exclude bots activities. We can see that the Arabic MML model (arRoBERTa$_{\text{BASE}}$) trained on no bot-generated articles scores better than the Arabic MLM model trained on bot-generated articles (20.41 and 23.70, respectively). Whereas in the case of the Moroccan Arabic MLM model (aryRoBERTa$_{\text{BASE}}$), we have opposite results, and we attribute that to removing the bot-generated articles from its corpora, making it even smaller.

| MLM MODEL | CORPORA | SAMPLES | PSEUDO-PERPLEXITY |
|---|---|---|---|
| **arRoBERTa$_{\text{BASE}}$** | With bots | 500 | 23.70 |
| | No bots | | 20.41 |
| **aryRoBERTa$_{\text{BASE}}$** | With bots | 500 | 5,379.89 |
| | No bots | | 5,686.44 |

Table 13: Pseudo-Perplexity scores of the Arabic Wikipedia and Moroccan Arabic Wikipedia MLM models before and after excluding the bot-generated articles.