bitext we help AI understand humans

Benchmark on Embeddings for Topic Modeling in Arabic

The impact of lemmatization for morphologically-rich languages

Introduction

Understanding text is one of the most important tasks that artificial intelligence algorithms are trying to solve. With the high-availability of textual data that lack structure or labels, text mining techniques, like topic modeling, became crucial. Topic models tries to summarize documents by extracting the most important topics in an unsupervised way. In this work, we investigate the impact of lemmatization on the performance of topic models.

Working with morphologically-rich languages, like Arabic or German, requires utilizing more complex text preprocessing techniques than the ones required when working with other languages with simpler morphology like English.

The complexity of morphology pose several challenges and not dealing with it properly can worsen the performance of language models. As an example of these complexities, the root word كتب "he wrote" in Arabic can be used to form more than 250 word forms like سيكتبون "They will write", مكتوب "written", and فكتبن "Then they (feminine) wrote". In addition, diacritics, which are not usually written in Modern Standard Arabic (MSA), can change the meaning of words, like كَتَبَ "he wrote" and كُتُب "books".

For this reason, leveraging a text normalization techniques, like lemmatization which returns different word forms to their base form, is important.

In this work, we worked with Arabic, which is one of the six U.N official languages and has a huge number of speakers globally. We worked with MSA which is the language that is used in news, education and literature throughout the Arab World.

We trained two variants of each of the following topic models: LDA, NMF, CTM, ETM, and BERTopic with non-lemmatized and lemmatized text. For BERTopic, we trained additional two variants that were initialized with custom word-based and lemma-based embeddings.

When lemmatizing text, we used the Bitext Lemmatizer. The Bitext Lemmatizer is high-performance platform-independent lemmatization library that can process millions of lookups per second. It provides support over 100 languages (77 languages and 25 variants) with high lexical coverage and comprehensive support for morphosyntactic features.

Methodology

1.Dataset:

We worked with NADiA1(1), an Arabic dataset that contains 35,416 articles which were extracted from the SkyNewsArabia Arabic news website(2).

The dataset was originally used for multi-label classification, where every article can belong to more than one category. In this work, we worked only with the articles and discarded the labels. We selected this dataset because we found it to be of high-quality and covers a diverse set of topics. It has the following 24 categories:

News, North Africa, Levant, Middle East, The Americas, Research, Finance & Economy, War & Terrorism, Gulf, Europe, Political Figures, Iran, Technology, Russia, Sports, Tennis, Football, English League, Arabian Sports, Spanish League, Health, East Asia, Environment, and Other Countries.

Preprocessing:

1- We filtered the articles to keep only the ones with a total word count of between 15 and 500 words. This is because working with very short or very long articles is out of the scope of our work.
2- We removed diacritics, which is a common preprocessing step when working with Arabic text.
3- We cleaned text by removing numbers, removing unnecessary whitespaces, and adding a space between punctuation marks and words.
4- We removed stopwords.
5- We lemmatized the text using Bitext lemmatizer.
6- We created a vocabulary using the most frequent 10,000 words in the corpus and processed the articles to keep only these words.

1. https://data.mendeley.com/datasets/hhrb7phdyx/2
2. https://www.skynewsarabia.com

2. Experiments:

For all of our experiments, we trained models with the following number of topics: 5, 10, 25, 50, 75, and 100. For each experiment, we trained a model for 5 runs and then averaged the obtained evaluation scores. We trained two variants of each model using both non-lemmatized and lemmatized text. For BERTopic, we trained addition two variants with word-based and lemma-based embeddings.

1.Latent Dirichlet Allocation (LDA):
We started by training LDA topic models. LDA is an unsupervised statistical technique that is commonly used for topic modeling.

For building models and training, we worked with the Gensim library(3).

We trained each model for 10 epochs. For the alpha hyperparameter, we chose the value of "auto" which enables the model to learn an asymmetric prior from the corpus.

2. Non-Negative Matrix factorization (NMF):
NMF is a multivariate analysis algorithm that leverages linear algebra to create topic models. We built and trained our NMF models using the Gensim library.

We trained each model for 30 epochs using gradient descent step size of 1.0.

3. Contextualized Topic Models (CTM)(4):
CTM leverages two representations of the text, pre-trained embeddings and bag-of-words representations.

To create contextualized embeddings, we used AraBERT v0.2(5). We used the raw text with minimum processing as the input of AraBERT.

For the bags-of-words model, we used the fully processed version of the text. We trained the CTM models for 10 epochs.

5. Embedded Topic models (ETM):
ETM is a technique for topic modeling that was introduced in 2019. It represents words and topics in the same embedding space and tries to leverage trained word embeddings in topic modeling.

3. https://radimrehurek.com/gensim
4. https://github.com/MilaNLProc/contextualized-topic-models
5. https://huggingface.co/aubmindlab/bert-base-arabertv02

Specifically, it was developed to deal with cases when there are large vocabularies. To train our ETM models, we used the OCTIS tool6). We initialized the models with custom embeddings that we trained on Wikipedia text. We will introduce more details on these embeddings in the next section.

6. BERTopic:

BERTopic is a novel topic modeling technique that was introduced recently(7). It creates topic models in four steps:

- It leverages a pre-trained language model, like BERT, to create embeddings of documents.
- The dimensionality of these embeddings are reduced using a dimensionality reduction algorithm like UMAP. This will help in the next step: the clustering.
- The embeddings are clustered into groups using a clustering algorithm like HDBSCAN.
- Finally, bag-of-words representations of clusters are created and class-based TF-IDF scores for each word are calculated. This will allow for selecting the most important words, namely the topics, from each cluster.

We trained 4 different versions of BERTopic using different pre-trained embeddings and different representations of text. Here is a summary of these four models:

BERTopic: We trained this model by initializing its embeddings using AraBERT v0.2 and using the non-lemmatized text. To obtain AraBERT, we leveraged the Flair framework(8).

BERTopic-lemma: This model was also initialized with AraBERT embeddings but we trained it on the lemmatized text.

BERTopic-lemma-word-embed: We initialized this model with custom word-level embeddings instead of AraBERT. The BERTopic tool supports initializing document embeddings by using word embeddings, where it calculates the mean of word vectors in each document. We started by training our own word embeddings on the latest dump of Arabic Wikipedia(9).

To do this, we cleaned Wikipedia text by removing numbers, punctuation, and non-Arabic words. After that, we trained the skip-gram model of Word2vec using a windows size of 5, a minimum word count of 5, and a vector dimension of 150. When training BERTopic, we used the non-lemmatized text.

6.https://github.com/MIND-Lab/OCTIS
7.https://github.com/MaartenGr/BERTopic
8.https://github.com/flairNLP/flair
9.https://dumps.wikimedia.org/arwiki/20230101

BERTopic-lemma-embed: We initialized this model with novel word-lemma embeddings. To train these embeddings, we used the word2vec_morph tool(10). Then, we trained BERTopic with the lemmatized text.

For the dimensionality reduction step, we used the default UMAP algorithm, considering that it gives reasonable performance. For clustering, we also used the default HDBSCAN algorithm.

Results and Discussion

We evaluated our models using the Topic Coherence (NPMI) metric on the test set that contains 3,416 articles.

Topic Coherence measures the interpretability of the generated topics by assessing the coherence of the top n words of each topic.

The resulting score is a decimal value between -1.0 and 1.0, where a higher score indicates more coherent topics.

Table 1 shows the resulting Topic Coherence scores of the different models.

10.https://github.com/pamelashapiro/word2vec_morph

	5	10	25	50	75	100
LDA LDA-lemma	**0.0856** 0.084	**0.0906** 0.0899	0.0943 **0.1073**	0.0757 **0.0931**	0.0293 **0.0626**	-0.0036 **0.0193**
NMF NMF-lemma	**0.1301** 0.0927	**0.1542** 0.1254	**0.1585** 0.1311	**0.1491** 0.1332	**0.1257** 0.1243	0.1092 **0.1118**
CTM CTM-lemma	-0.0134 **0.0042**	0.0643 **0.0795**	0.0774 **0.087**	0.0787 **0.0889**	0.071 **0.0812**	0.0702 **0.0748**
ETM ETM-lemma	-0.0889 **-0.0328**	-0.0545 **-0.0181**	-0.0337 **0.0005**	-0.0619 **-0.0122**	-0.0787 **-0.0333**	-0.0974 **-0.0482**
BERTopic BERTopic-Lemma	0.1628 **0.1647**	0.1596 0.1699	0.1565 0.1692	0.1254 0.1409	0.0845 0.1207	0.0598 0.0965
BERTopic- word-embed	0.1249	0.1511	0.1544	0.1479	0.1131	0.0852
BERTopic-lemma-embed	0.1518	**0.1754**	**0.1726**	**0.1569**	**0.1425**	**0.1241**

Table 1 Topic Coherence scores of our models with different numbers of topics. Each score is calculated by averaging the resulting scores of 5 runs

The results show that, on most cases, training topic models on lemmatized text leads to better performance. In addition, we see that initializing BERTopic with lemma-based embeddings leads to better performance than using either AraBERT or word-level embeddings.

This highlights the importance of lemmatization to normalize text when working with languages with complex morphology like Arabic.

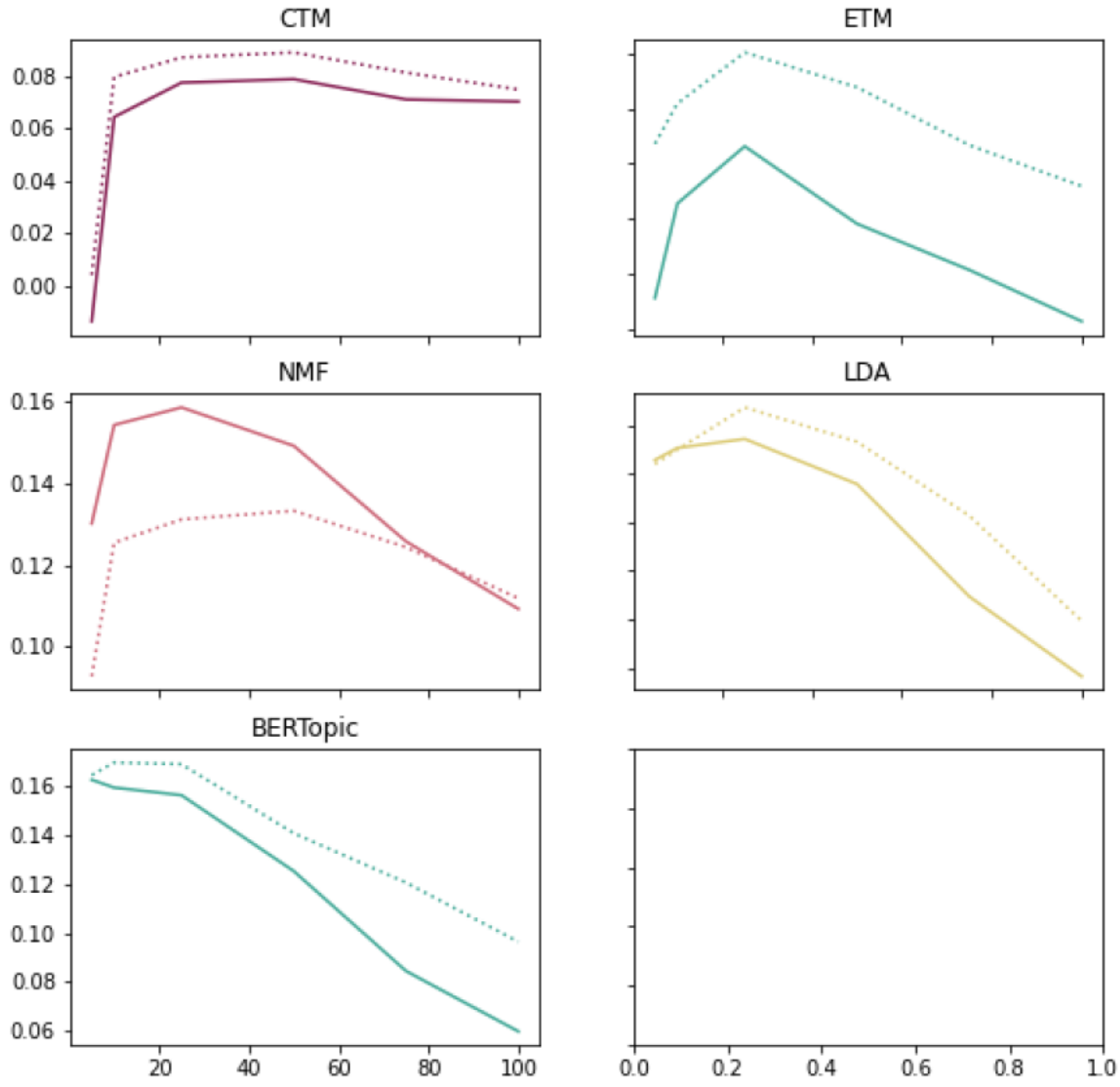

Figure 1 A comparison of Topic Coherence scores of different models. The dotted lines refer to models trained on lemmatized text, whereas the solid lines refer to models trained on non-lemmatized text

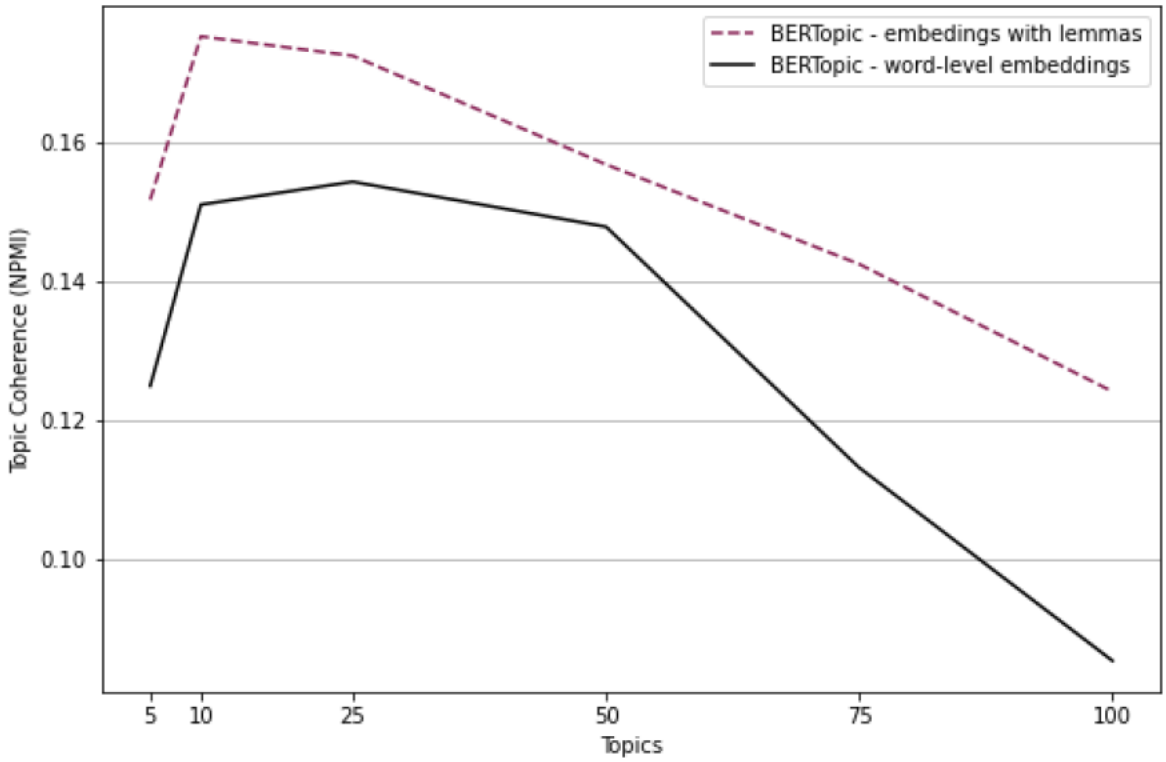

Figure 2 A comparison of the Topic Coherence scores of the two variants of BERTopic that were trained using word-level and lemma-based embeddings

Conclusion

In this work, we investigated the effects of leveraging Bitext lemmatizer for the task of topic modeling in Arabic. We worked with a high-quality Arabic dataset of news articles to train five models: LDA, NMF, BERTopic, ETM, and CTM. For the BERTopic model, we trained 4 variants using both AraBERT and Word2vec word and lemma-based embeddings that we trained on Wikipedia text.

We evaluated our models on a separate test set using the Topic Coherence (NPMI) metric. Our results show that applying Bitext lemmatizer on text yields better topic models and higher Topic Coherence scores, with the exception of NMF models. Also, we showed that using lemma-based embeddings when initializing BERTopic leads to better performance than using word-level embeddings.

Select Customers