



**UNIVERSITÀ
DEGLI STUDI
DI TRIESTE**



DATA SCIENCE &
SCIENTIFIC COMPUTING

Natural Language Processing

Multilingual NLP: A comparative study of monolingual BERT and multilingual BERT

Fatma Moustafa

June 29, 2023



- Introduction
- Literature Review
- Architecture of models used
- Pretraining
- Finetuning
- Dataset and NLP task description
- Test cases
- Results and Conclusion



Introduction

Why multilingual NLP is important?

- World is multilingual. NLP should cater for such diversity.
- Around 7000 languages with thousands of variations. Each has its own rules, scripts, symbols and structure.
- Different language → different representations → different language models
- Most models are English based or European based.
- 94% of world's population first language is not English. 74% of the world's population do not speak English at all.
- Majority of the world's population in Africa, Asia and Americas speak languages that are not presented in NLP models or with very few resources.

- **Solution:**

- ① Monolingual specific models:

- Does not scale well
 - Training new models is costly

- ② Multilingual models:

- Robust multilingual representations
 - Curse of multilinguality, only 100's of languages represented from 7000
 - Lack of pre-training data

- ③ Translation:

- Less expensive than training dedicated language specific model
 - Overall response time increases

Multilingual BERT (mBERT)

- Single language model pre-trained from 104 most common languages in Wikipedia (monolingual corpora)
- High/ Medium/ Low resource languages: Size of Wikipedia of the language
- **Advantages**
 - Zero-shot Cross-lingual model transfer: Train the model in one language (high resource like English) and fine tune it for tasks in another language (could be low resource language)
 - Lexical overlap between languages improves transfer.
 - Captures multilingual representations
 - Transfer between languages even if: low lexical overlap OR different scripts

- NOTE: Most evaluations on cross-lingual transfer focused on only a third of the languages covered by mBERT (mostly high resource).
- **Why mBERT might perform worse?**
 - Lack of data → Unable to learn high quality representations in low resources languages
 - Languages resources also vary in size.
Solution: upsample low resourced words and downsample high resourced words
 - Low corpus size, low quality representation and non uniform vocabulary distribution in mBERT
- NOTE: Correlation between mBERT performance and resources does not explain causality of bad performance



Literature Review

Paper 1: How multilingual is Multilingual BERT?

Experiments:

- ① Task: Name Entity Recognition(NER)
Dataset: CoNLL datasets
Number of languages: 16 languages
Focus: Western languages pairs
Result: Around 65% accuracy
- ② Task: Part of speech(POS) tagging on Universal Dependencies(UD)
Number of languages: 41 languages
Result: Around 80% accuracy
- ③ Task: Transliterated text formats
Result: mBERT was not efficient in transferring to transliterated target



④ Task: Translation

Dataset: WMT16 dataset(5000 pairs of sentences were sampled)

Results: Accuracy depends on the different pairs

Paper 2: Are All Languages Created Equal in Multilingual BERT?

- More languages were investigated in this paper.
- Experiments:
 - ① Task: NER
Number of languages: 99 languages
Results: For low resourced language, mBERT perform worse than a non BERT model
For low resourced language, multiple monolinigual models does not enhance performance either.
For high resource language, mBERT performs worse.
 - ② Task: POS tagging
Number of languages: 54 languages
Results: No notable performance gain using mBERT over BERT



- ③ Task: Dependency Parsing
Number of languages: 54 languages
Results: Same as POS tagging

Paper 3: Should we Stop Training More Monolingual Models, and Simply Use Machine Translation Instead?

- IDEA: Translate data into English.
Use pre-trained large-scale English Language Models.
- English LM improved tremendously, thus; translation improved as well
- Empirical and environmental stand-point is more effective to translate data from low resource language than build LM for these languages.
- There are more high quality translation models than there exists training data for building LM.



- Experiments:
 - ① Task: Sentiment Analysis (SA)
Focus: Scandinavian Languages
Result: Machine translation + English LM perform better than monolingual models in most cases.



Architecture of models used



BERT

- BERT: Bidirectional Encoder Representation of Transformers
- It has two distinct phases:
 - ① A pre-training phase
 - ② A fine-tuning phase
- Bidirectionality is injected into the pre-training phase.
- INPUT: Sequence (sentence or pair of sentences)
- CLS: First token of sentence
CLS final hidden vector: classification tasks
- SEP: Token to separate sentences

Architecture of models used II

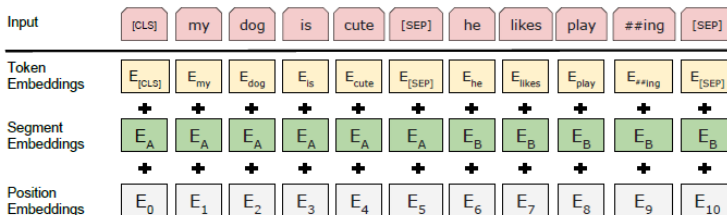


Figure: BERT embedding layer

- EMBEDDINGS:

- ❶ **Token embedding:** Pre-trained embeddings for different words
- ❷ **Segment embedding:** Which sentence each token belongs to
- ❸ **Position embedding:** Absolute position of word in sentence



Pre-training

BERT

- Datasets and vocabulary size:
 - ① BooksCorpus -800M words
 - ② English Wikipedia -2,500M words
- TASKs:
 - ① Masked Language Modeling (MLM)
 - Allows for bidirectionality
 - 15% of tokens randomly selected:
 - 80% replaced with [MASK] Token
 - 10% Replaced by random token
 - 10% left unchanged
 - ② Next Sentence Prediction (NSP)
 - Understand relation between sentences
 - Given pair of sentences A and B
 - Learn if B succeeds A



Finetuning



BERT

- BERT model is initialized with the pre-trained parameters
- All parameters are fine-tuned using labeled data from the downstream tasks.
- A fully connected layer is added on top of BERT and trained for a few epochs
- For the output layer a [CLS] representation for classification like sentiment analysis tasks



AraBERT

- Arabic: morphological rich language with few resources and not enough exploration compared to English
- Arabic pre-trained LM based on google's BERT architecture
- Same BERT Base configuration and same pretraining steps
- Dataset: 1.5 billion words Arabic Corpus and 3.5 million articles from Open Source International Arabic News Corpus
- AraBERT was applied to three tasks of Natural Language Understanding(NLU): Sentiment Analysis(SA), Named Entity Recognition(NER) and Question Answering(QA).
- Results of AraBERT were compared to that of mBERT, showing that AraBERT outperformed mBERT.



AIBERTO

- Italian LM focusing on the language used in social networks, especially Twitter
- Trained only on MLM task not NSP. Tweets are not dialogues.
- Not suitable for Question Answering task
- Dataset: TWITA corpus having 200 million tweets
- AIBERTO was applied to three sub-tasks: Subjectivity Classification, Polarity Classification and Irony Detection
- AIBERTO performance was better than the multilingual LM.



Dataset and NLP task description

- The ("*cardiffnlp/tweet_sentiment_multilingual*") dataset from HuggingFace was used.
- It is made up of tweets from Twitter in 8 languages: English, Arabic, Italian, German, French, Portuguese, Hindi and Spanish.
- Each tweet text is associated with a class label(sentiment).
- Classes are: 0 = negative, 1 = neutral and 2 = positive.
- The dataset is balanced.
- Each language has 3033 rows that are divided to training set (1839 rows), validation set(324 rows) and test set (870 rows).
- The task intended for this dataset is text classification: Sentiment Analysis.



Test Cases

- Performance of monolingual BERT and its multilingual counterpart were studied. Sentiment Analysis was done in English, Arabic and Italian languages.
- For the monolingual case, BERT and 2 language specific variants from the HuggingFace library were applied:
 - ❶ BERT: The vanilla BERT
 - ❷ AraBERT: Arabic variant of BERT
(*"aubmindlab/bert-base-arabert"*)
 - ❸ ALBERTo Italian variant of BERT (*"m-polignano-uniba/bert_uncased_L-12_H-768_A-12_italian_alb3rt0"*)

- **Common Preprocessing tasks:**

- ➊ Remove punctuations, stopwords, links, mentions and new line characters.
- ➋ Clean hashtags at the end of the sentence, and keep those in the middle of the sentence by removing just the # symbol.
- ➌ Filter special characters such as & and \$ present in some words.
- ➍ Remove multiple spaces.
- ➎ Expand contractions.
- ➏ Remove numbers.
- ➐ Remove short words
- ➑ Replace elongated words with their base form.
- ➒ Remove repeated punctuation.
- ➓ Remove extra whitespace.
- ➔ Remove short tweets.
- ➕ Remove URLs



13 Remove Emojis



- **English-specific Preprocessing tasks:**
 - ① Lemmatize words: Use WordNetLemmatizer from nltk library.
 - ② Stemming: Use PorterStemmer from nltk library.
 - ③ For stop words removal, English was specified.

- **Arabic-specific Preprocessing tasks:**

- ① Normalize words: substitution of some different scriptures of a single letter.
- ② Normalize words: Removal of Tashkeel, Tanwin and Madd (similar to accents) using pyarabic.araby library
- ③ Tashkeel: Each letter has 4 sounds: silent, A, U, I sound that affects pronunciation of words, thus, completely changing the meaning.
- ④ Tanween: Letters are stressed with a certain Tashkeel sound (not silent) and an N end. Tanween does not change the meaning of the word.
- ⑤ Madd: Used with vowels to elongate its sound: 1 second for vowel and 5-6 seconds for Madd. It does not change the meaning of the word.
- ⑥ Lemmatization and Stemming: Use ArabicLightStemmer from tashaphyne.stemming library.



- ⑦ Letters are also separated since letters in each word are connected.
- ⑧ For stop words removal, Arabic was specified.
- ⑨ **farasapy** library was also used to perform a clean text preprocessing.

- Italian-specific Preprocessing tasks:

- ① Lemmatize words: simplelemma library was used. Italian language was specified.
- ② Stemming: SnowballStemmer('italian') was used from nltk.stem library.
- ③ For stop words removal, Italian was specified.
- ④ **pretwita** library was also used to perform clean text preprocessing.

- After cleaning the tweets, they are tokenized to get input IDs and attention masks.
- Tokenized tweets are encoded to get the maximum length of the training tweets.
- Create DataLoaders to be used in training the model.
- Finetuning: A custom BERT classifier is created from original model and additional Dense layers are added to perform the classification task.
- During training, 10 epochs were used. Average training loss, validation loss and validation accuracy were calculated.
- Predictions were made on the test set. Classification Report and a confusion matrix showed the results of the prediction.



- More experiments were done initially, but only final and optimal case solutions were portrayed.



Results and Conclusion



	precision	recall	f1-score	support
Negative	0.60	0.76	0.67	286
Neutral	0.51	0.51	0.51	277
Positive	0.76	0.55	0.64	282
accuracy			0.61	845
macro avg	0.62	0.61	0.61	845
weighted avg	0.62	0.61	0.61	845

a) Classification Report

BERT Sentiment Analysis
Confusion Matrix

Test	Predicted		
	Negative	Neutral	Positive
Negative	218	52	16
Neutral	103	140	34
Positive	41	85	156

b) Confusion Matrix

Figure: BERT used for English

Results II

	precision	recall	f1-score	support
Negative	0.58	0.70	0.64	284
Neutral	0.48	0.54	0.51	274
Positive	0.76	0.50	0.60	282
accuracy			0.58	840
macro avg	0.61	0.58	0.58	840
weighted avg	0.61	0.58	0.58	840

a) Classification Report

BERT Sentiment Analysis
Confusion Matrix

Test	Predicted		
	Negative	Neutral	Positive
Negative	199	70	15
Neutral	95	149	30
Positive	47	94	141

b) Confusion Matrix

Figure: mBERT used for English

Results III

	precision	recall	f1-score	support
Negative	0.88	0.58	0.70	290
Neutral	0.50	0.74	0.60	290
Positive	0.74	0.64	0.69	290
accuracy			0.65	870
macro avg	0.71	0.65	0.66	870
weighted avg	0.71	0.65	0.66	870

a) Classification Report

araBERT Sentiment Analysis Confusion Matrix				
Test	Negative	167	112	11
	Neutral	20	216	54
	Positive	3	101	186
		Negative	Neutral	Positive

b) Confusion Matrix

Figure: AraBERT used for Arabic

Results IV

Classification Report for araBERT :

	precision	recall	f1-score	support
Negative	0.42	0.56	0.48	290
Neutral	0.33	0.47	0.39	290
Positive	0.46	0.12	0.20	290
accuracy			0.38	870
macro avg	0.40	0.38	0.35	870
weighted avg	0.40	0.38	0.35	870

a) Classification Report

araBERT Sentiment Analysis
Confusion Matrix

Test	Predicted		
	Negative	Neutral	Positive
Negative	161	106	23
Neutral	136	135	19
Positive	88	166	36

b) Confusion Matrix

Figure: mBERT used for Arabic

Results V



	precision	recall	f1-score	support
Negative	0.75	0.44	0.56	288
Neutral	0.58	0.72	0.64	285
Positive	0.55	0.64	0.59	290
accuracy			0.60	863
macro avg	0.63	0.60	0.60	863
weighted avg	0.63	0.60	0.60	863

a) Classification Report

BERT Sentiment Analysis Confusion Matrix			
Test	Negative	Neutral	Positive
	127	70	91
	18	204	63
Positive	24	79	187
	Negative	Neutral	Positive
Predicted			

b) Confusion Matrix

Figure: ALBERTo used for Italian



	precision	recall	f1-score	support
Negative	0.65	0.32	0.43	288
Neutral	0.52	0.72	0.60	285
Positive	0.49	0.56	0.52	290
accuracy			0.53	863
macro avg	0.56	0.53	0.52	863
weighted avg	0.56	0.53	0.52	863

a) Classification Report

**BERT Sentiment Analysis
Confusion Matrix**

Test	Negative	92	91	105
	Neutral	19	206	60
	Positive	30	99	161
		Negative	Neutral	Positive
		Predicted		





b) Confusion Matrix





Figure: mBERT used for Italian


Table: Comparison of Accuracy for monolingual and multilingual BERT LM for the 3 languages

Language	LM	Test accuracy %
English	BERT	61
English	mBERT	58
Arabic	AraBERT	65
Arabic	mBERT	38
Italian	AlBERTo	60
Italian	mBERT	53

- Monolingual models: AraBERT outperformed BERT and ALBERTo by a small gap.
- Multilingual model: Using multilingual BERT for Arabic had the least performance. The best performance was for English.
- The prediction of classes were distributed uniformly. No bias to one class.
- mBERT might be able to learn multilingual representations, but it does not outperform BERT and its language specific variants.
- For a language like Arabic which is low resource and has a very different lexical and morphological format than English and Italian, it is better to train specific language models.

- 
-  Antoun, Wissam, Fady Baly, and Hazem Hajj (2020). “Arabert: Transformer-based model for arabic language understanding”. In: *arXiv preprint arXiv:2003.00104*.
 -  Devlin, Jacob et al. (2018). “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805*.
 -  Isbister, Tim, Fredrik Carlsson, and Magnus Sahlgren (2021). “Should we Stop Training More Monolingual Models, and Simply Use Machine Translation Instead?” In: *arXiv preprint arXiv:2104.10441*.
 -  Pires, Telmo, Eva Schlinger, and Dan Garrette (2019). “How multilingual is multilingual BERT?” In: *arXiv preprint arXiv:1906.01502*.

-  Polignano, Marco et al. (2019). “Alberto: Italian BERT language understanding model for NLP challenging tasks based on tweets”. In: *CEUR Workshop Proceedings*. Vol. 2481. CEUR, pp. 1–6.
-  Ruder, Sebastian (2022). *The State of Multilingual AI*. <http://ruder.io/state-of-multilingual-ai/>.
-  Vargas Feijó, Diego de and Viviane Pereira Moreira (2007). “Mono vs multilingual transformer-based models: a comparison across several language tasks”. In: *CoRR*, *abs*.
-  Wu, Shijie and Mark Dredze (2020). “Are all languages created equal in multilingual BERT?” In: *arXiv preprint arXiv:2005.09093*.



Thank you for your kind attention