NLP Project Report

Think You Know English? Think Again...

Group 9
Alina Gavrish, Dominic Sagers, Léa Hiebel, Sofia Ozhogina, Tantus Choomphupan

May 2023

1 Abstract

This paper attempts to tackle the problem of native English vs. non-native English recognition. Using a data-set containing passages written by native and non-native speakers, different classification models were tested to classify them. The BERT (Bidirectional Encoder Representations from Transformers) provided the highest performance, and was further optimized. Through our research we discovered that BERT performs extremely well on the classification of native and non-native speakers given a relatively small amount of data (less than 15,000 instances).

2 Introduction

Native Language Identification (**NLI**) is a well-established problem in Natural Language Processing (**NLP**), where the goal is to identify a writer's native language from his/her writing in a second language, usually English. For this work, we focused on identifying whether the writer is a native or not English native speaker because it is one of the crucial first steps to predict from which part of the world a speaker is. This knowledge can also be applied for voice assistants, which can struggle depending on the dialect of English of the speaker. A potential next step of this project could be to identify precisely what are differences between native and non-natives. This could be used to have voice assistants trained on a specific dialect of English, or with a specific accent/vocabulary.

We set out to see if, using state-of-the-art architecture, we could make a classification model which can confidently predict whether or not a given passage of English text was written by a native speaker or not. Then, the main focus of the paper is shifted to identifying the possible reasons for the existing misclassifications and improving the performance by taking them into consideration.

We then chose to focus on two research questions:

> *RQ1: How well does the state-of-the-art classification model BERT classify native and non-native speakers?*
>
> *RQ2: What are the possible reasons behind misclassifications?*

This paper's structure is as follows: We first begin by describing the related works which inspired the direction in developing our model. Then in the section "Dataset" we describe the data set used to train our model. In the section "Methods", a detailed account of which algorithms and practices were used in creating our model are given so that our work is replicable. Then in "Experiments and Results" we describe the experiments performed and their respective results. We finish with the "Discussion" and "Conclusion" section, which encapsulate the overall work.

3 Related work

The chosen methods are inspired by the related research of similar problems. Similar researches are tackling problems such as native language identification on English text written by native speakers of

English, Korean, Chinese and Japanese[2]. In addition, we also looked into a paper that aims at evaluating levels of English based on spontaneous and non-spontaneous English texts provided by participants and extracted from the web.

4 Dataset

Our dataset is comprised of a combination of two labeled datasets which comprised 13,617 (7902 native/5,717 non-native) labeled instances of native and non-native texts.

4.1 Kaggle Dataset

The initial dataset used in the project is an open-sourced native vs non-native text corpus from Kaggle [5] (an online, open-source dataset aggregator). The dataset contains two sets of English texts: one generated by a group of non-native authors, while the other is from English-speaking news outlets like the BBC and the New York Times and is labeled as native.

During our testing, this dataset proved to be too small (561 entries), thus a more comprehensive dataset was needed as surplus data. This dataset will be referred to as "short dataset".

4.2 Academic Essay Dataset

Additionally, we also combined two datasets from ICNALE [12] - corpus containing essays by Asian students, and LOCNESS [4] - corpus containing essays by native students. Both corpora were produced by students with similar academic levels, for ICNALE it was a university-entrance English test, while LOCNESS is A-level and British/American university essays. This dataset will be referred to as "long dataset".

5 Methods

This section will describe the necessary knowledge required to interpret, understand, and recreate the approaches used in later sections.

5.1 Pre-processing

We performed the pre-processing of the text corpus by using the regular expression library provided by NLTK [1] to get rid of the abbreviations and non-alphabetic characters, however, we decided to keep the casing of the words, in case it yields valuable insight for the texts' author. We also chose to keep certain symbols that occur frequently such as hyphens ("-") and forward slashes ("/"), as removing them created issues wherein words would be concatenated. Removing these characters would also have a possible effect on classification as perhaps natives or non-natives would use them differently. The result is a corpus which is pre-processed to preserve both syntactic and semantic meaning while also remaining within the bounds of the English language.

5.2 Input Encoding: BERT Embeddings

To turn our training text sentences into an input for classification, we used a recently popular approach of sequence embedding: a word vector embedding based on the Bidirectional Encoder Representation from Transformers (**BERT**) architecture (provided by SBERT.net[10]).

BERT is a bidirectional transformer encoder, which comes already equipped with pre-trained weights for word embeddings. We used it further for fine-tuning embeddings using our own training set corpus. The variation of model used in our research (bert-base-nli-mean-tokens) generates sentence embeddings by taking the mean of the word embedding produced by BERT.

One of the main advantages of the BERT model is its use of bi-directionality when creating embeddings, meaning that it considers the context of a word by looking at both its preceding and following words. It enables the model to have a deeper sense of language context and have improved many natural language processing tasks. [3] We believe it is able to capture the difference between native and non-native English speakers and this conclusion was borne out by our research.

5.3 Local explanation using LIME

LIME [11] (Local Interpretable Model-agnostic Explanations) is a library for generating local explanations of machine learning models, including those applied to NLP tasks. It provides a framework to understand the predictions of black-box models by approximating their behavior with interpretable models on local instances. LIME aims to answer the question of why a particular model made a specific decision on a given instance. LIME works by generating perturbations or "perturbed instances" of the input text and observing the changes in the model's predictions. It constructs an interpretable model, such as linear regression or decision trees, on these perturbed instances to approximate the behavior of the black-box model in the local neighborhood of the input.
The way we use LIME in our code is by passing it the misclassified instances of our classifier. This allows us for a local explanation of why an instance was misclassified. In the experiment section, we will detail our methodology some more and explain the format of the output given by LIME.

5.4 Classifiers tested

5.4.1 SVM

For the classifier, we used SVM model with radial basis function kernel provided by scikit-learn library [9]. We also performed a grid search on the model parameters such as kernel type, penalty strength coefficient, and penalty type.

5.4.2 Naive Bayes

Naive Bayes is based on the idea of maximum likelihood, trying to make a prediction using a product of the conditional probability of each word in the context. Additionally, we are using the Gaussian implementation of the algorithm as proposed by Stanford working paper [13], this version assumes that the data follows the Gaussian distribution and can perform very efficient calculations for the classifier. In NLP, the task is computing the conditional probability of the native/non-native class given the sentences' embedding.

5.4.3 Logistic Regression

Unlike the Naive Bayes, Logistics Regression(LR) is a discriminative classifier that tries to learn the difference between classes. To find the relationship between the features and the outcome, LR calculates the weighted sum of the features, where each feature is multiplied by its corresponding weight. After getting the data, the Sigmoid function is used to map predicted values with their probabilities [7].

6 Experiments & Results

In this section, we present the experiments conducted. Our aim is to understand why some instances of our dataset were misclassified, while others were not.

6.1 Hyperparameters Tested

We performed a grid search to discover which hyperparameters for our SVM classifier resulted in the best performance using our embeddings. The relevant parameters we searched were the kernel type and the regularization constant strength "C".

During our testing, we found that a kernel type of "rbf" and a regularization constant strength of C=8 proved the best combination of parameters.

6.2 Comparison in classifiers

For SVM, we tested for the kernels "linear", "rbf" and "sigmoid" and for values for C between 1 and 10. For Logistic regression, we tested for penalties of "l1" or "l2" and regularizations of 1 to 10. Finally, for Naive Bayes, we tested values for alpha smoothing between 1 and 6. The accuracy for the best set of parameters are presented in the following table :

Data	SVM	Logistic Regression	Naive Bayes
small dataset	0.9458	0.926	0.8272
big dataset	0.9822	0.9572	0.8536

Table 1: Results on test data

The results are the performance on our testing set, with the parameters for each model presented in the sections before.

7 Discussion

Below we apply our interpretations of the results from our experiments.

7.1 Performance of Classifiers

According to Ng and Jordan, the Naive Bayes performs better on small-scale data [8]. However, according to our results in Section 6.2, the LR outperforms Naive Bayes due to the fact that LR is better at classifying the correlated data. In the case of native/non-native data, there exists a correlation in terms of the sentence structure, thus LR assigns a better probability compared to Naive Bayes. [6]. Alternatively, we can see that SVM classifies performed the best due to the fact that SVM is based on the geometrical properties whereas the LR is based on a statistical approach. When it comes to overfitting the data, the risk is less in SVM compared to LR and Naive Bayes.

7.2 Analyzing the Test Set Data

Our model performs very well on the supplied dataset, though it helps to understand where our model failed so that further improvements can be made.

7.2.1 Potential errors in original dataset

Analyzing the results and seeing instances which were misclassified most egregiously (figures of which can be found in the appendix under the section "LIME Results: Outliers"), we can see that our model was very confident in one classification, and applied very high weights to every word to determine its decision, but, in fact, the instance was of the opposite class.

We can assume given the high performance of our model and the surplus of data provided in training, that our model encountered a sentence which was spoken by a non-native/native speaker who was actually writing in a complementary way. Therefore, although it is not impossible that a native speaker might speak non-natively a small amount of time or vice-versa, these can be considered outliers and a fault in the data due to the confidence of our model.

7.2.2 Local errors with LIME

The local errors (instances with an incorrect classification wherein the model was not confident in either either class) are more difficult to interpret and explain.

Figures in the Appendix under "LIME Results: Misclassified Instances" display examples of such errors. In cases such as seen in **??**, our model applied a very low percentage confidence about the identity of the class in every word in a given instance (less than 0.5 on average), and chose the wrong class. Although the prediction probability of the incorrect class is extremely high, this is not representative of the model's overall confidence, for the miniscule weighting for each word guarantees a classification that is closer to random as the model is not confident at all.

In other cases our model made a legitimate error (such as in Figure 2 and Figure 3) which was based on reasonable weighting and gave a confident incorrect decision, and this is less explainable, but most likely the.

It is possible that these errors could be amended by adding more training data, as the most likely cause of these misclassifications is a lack of experience with the occurrence of such problematic instances.

8 Conclusion

Through our research we discovered that BERT performs extremely well on the classification of native and non-native speakers given a relatively small amount of data (less than 15,000 instances). This is unsurprising as the architecture of the BERT encoder combines the best achievements of modern NLP.

Although there weren't many, the missclassifications which occured were observed to be mostly edge-cases where non-natives or natives might have hap-

pened to speak in a way which may be reminiscent to another, but this is rare. It is possible that these missclassifications can be accounted for with an increase of size in our dataset, or more fine-tuning of BERT. This is grounds for possible future research.

To be able to classify non-native and native English speakers to such a high degree of accuracy with a relatively small amount is a testament to how far the field of NLP has come in the last decade, and there is no indication this exponential progress will not continue in the future.

9 Appendix

9.1 LIME Results

9.1.1 Misclassified Instances

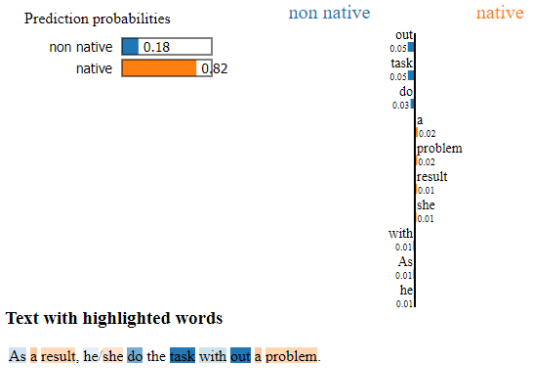

Figure 1: An example of a misclassified instance.

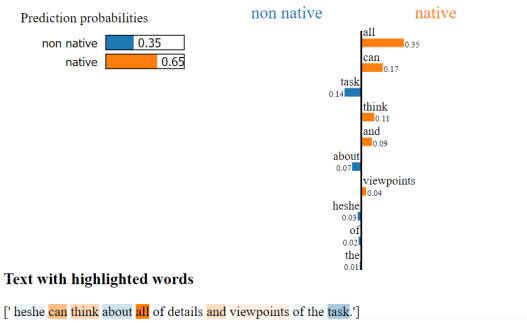

Figure 2: An example of a misclassified instance.

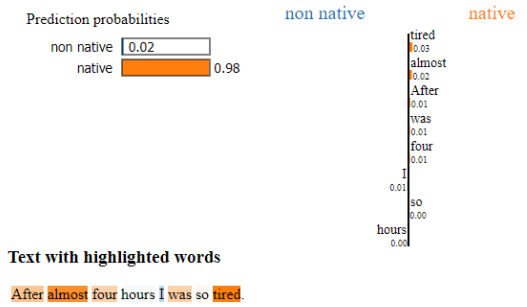

Figure 3: An example of a misclassified instance.

9.1.2 Outliers

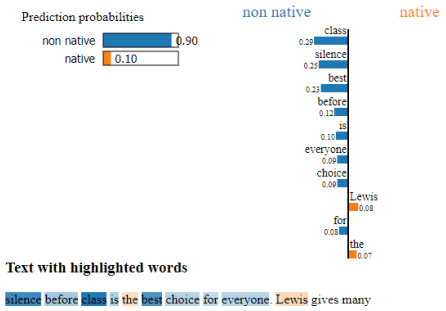

Figure 4: Example of a confidently misclassified native instance.

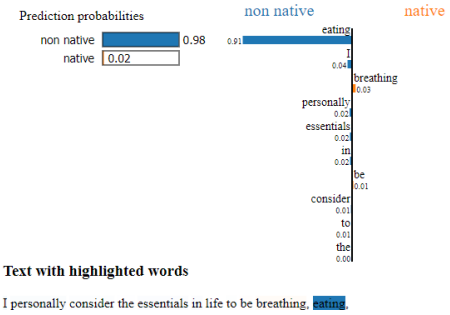

Figure 5: Example of a confidently misclassified native instance.

References

[1] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[2] Jake Browning. "Using Machine Learning Techniques to Identify the Native Language of an English User". In: (2017).

[3] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[4] Centre for English Corpus Linguistics (CECL). *The Louvain Corpus of Native English Essays.* Universit e catholique de Louvain, Belgium, 2023.

[5] Ahmadali Jamali. *Native or Non-native Dataset.* `https://www.kaggle.com/datasets/ahmadalijamali/native-or-non-native`, 2023.

[6] D. Jurafsky et al. *Speech and Language Processing.* Pearson Education, 2014. ISBN: 9780133252934. URL: `https://books.google.nl/books?id=Cq2gBwAAQBAJ`.

[7] Geoffrey Morrison. "Logistic regression modelling for first and second language perception data". In: Jan. 2007, pp. 219–236. DOI: `10.1075/cilt.282.15mor`.

[8] Andrew Ng and Michael Jordan. "On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes". In: *Advances in Neural Information Processing Systems.* Ed. by T. Dietterich, S. Becker, and Z. Ghahramani. Vol. 14. MIT Press, 2001. URL: `https://proceedings.neurips.cc/paper_files/paper/2001/file/7b7a53e239400a13bd6be6c91c4f6c4e-Paper.pdf`.

[9] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[10] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: Association for Computational Linguistics, 2019. DOI: `\url{https://arxiv.org/abs/1908.10084}`.

[11] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining.* 2016, pp. 1135–1144.

[12] S. Shikawa. *ICNALE Written Essays Dataset.* Kobe University, Japan, 2013.

[13] G.H. Golub R.J. LeVeque T.F. Chan. "Updating formulae and an pairwise algorithm for computing sample variances". In: (1979).