Automatic Identification of Related Languages

Natallia Casey

Spring Semester 2021

https://natacasey.github.io/

INTRODUCTION

Businesses and research communities around the world understand and value multilingualism more with the current acceleration of globalization. English, German, Spanish, and French are on the list of the most popular languages in higher demand for business (Pimentel, I.). Learning how to identify and analyze the text in these languages can provide a competitive advantage.

To bring value to multilingual research and to create a tool that could assist in multilingual preprocessing tasks, the project focuses on designing a language identifier. A language identifier is a tool that is capable of recognizing the natural language of the text. Unique and informative linguistic features of each of the related languages mentioned above, two different ways to preprocess data, and several supervised models such as Multinomial Naïve Bayes, Logistic Regression, Random Forest, Support Vector Machines with the one-vs-one scheme, Support Vector Machines with the one-vs-rest scheme, and a Bidirectional LSTM are used in this project to identify the best language identification workflow. The best performing model, cross-validated Multinomial Naïve Bayes with the accuracy of 99.994% using the data preprocessed with preserving unique linguistic peculiarities of the languages, and a customized tf-idf with the word-level n-grams as large as trigrams constitute the pipeline capable of successfully identifying the related languages even on short strings of both formal and informal text domains. The performance of the created language identification tool is also compared to the existing language detection tools showing promising results.

Background of the problem

Essentially, language identification is a text classification problem. Yet, it is very unique since some of the text preprocessing techniques used for the English language cannot be applied to other languages.

Near-perfect accuracy has been achieved by several existing language identification tools when working with long text strings. The language identification task becomes much harder when the text is short.

Language identification tools are known to do well for the domain data they have been trained on. Creating a tool that can generalize well outside the domain is an important but challenging task.

This project focuses on just 4 languages: English, French, German and Spanish. Even though it is a small number of languages the difficulty lies in the fact that they belong to the same proto-language (ancestor), Indo-European. Thus, a lot of the words in these languages have the same or similar word stems and similar linguistic characteristics.

To be able to differentiate between the related languages, the project stresses the importance of preprocessing techniques relying on the linguistic peculiarities of each of the languages. The differences in morphology and punctuation represent the core of developing the text preprocessing approach in this project. The efficiency of such an approach is seen in the results produced by the same models but with the data preprocessed in two different ways:

- accounting for linguistic nuances
- preprocessed following the steps similar to a monolingual text classification problem.

PREVIOUS RESEARCH

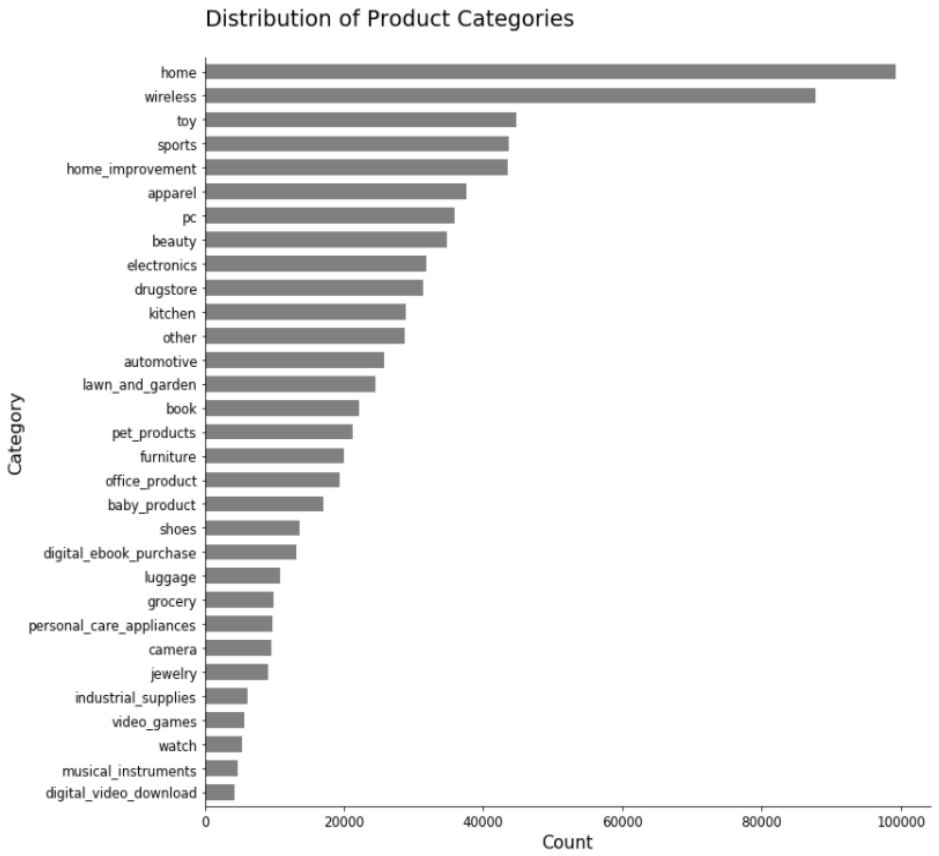
Existing approaches and techniques in the related research aimed at solving various challenges in language identification: cross-domain accuracy, accuracy on short text, and accurate identification of related languages. For example, M. Lui and T. Baldwin trained the byte-level n-grams of the data from 5 different domains with the Multinomial Naive Bayes and were successful at creating a high-performing language identification tool (Langid.py) which we also test in this paper. P. Mathur., A. Misra, and E. Budur created a language identification classifier using an ensemble of Recurrent Neural Networks with an accuracy of 95.12% for similar languages. They also tested the performance of Naïve-Bayes, Regularized Logistic Regression with char-level and word-level n-grams without any preprocessing. A bi-

directional LSTM was suggested as a promising means to improve short text language classification results by the engineers from Apple.

Some of the tools that resulted from the above-mentioned research are tested on the long and short strings of the formal and informal language style text data alongside the models created in this project to compare their performances.

DATA

FIGURE 1 | Product Categories of Reviews



The data set used for this project is the "Multilingual Amazon Reviews Corpus" with the selected languages of English, German, French, and Spanish. The reviews present in the data set were gathered in the time frame of November 1st, 2015 to November 1st, 2019 (Keung, P., Lu, Y., Szarvas, G., and Smith, N. A.). This is a balanced data set containing 820,000 records. The data set contains 8 variables. The variable "language" is a target variable in this analysis, and the variable "review_body" is a predictor variable. The reviews were given to many product categories as seen in **Figure 1.** 31 product categories used in the data set are indicative of a wide range of vocabulary present in the corpus. The length of the reviews in the "review_body" variable ranges from a minimum of 20 characters to a maximum of 2,000 characters. All the data is anonymized.

80 balanced observations were excluded from the testing and training data and shortened to the 3-word strings to test the models' performance on short text.

Another sample consisting of 80 balanced observations was created from version 6 of the European Parliament Proceedings Parallel Corpus. It is a corpus that contains the proceedings of the European Parliament and covers the languages of English, Spanish, German, and French. The style of the text is formal. It is only used to test the models' performance to see how well the models work outside of the domain they were trained on. A shortened version (3-word strings) of this sample is also used to explore the limitations and see how the resulting models handle short samples outside of the domain.

LINGUISTIC ANALYSIS

The languages used in this project are related and belong to the same proto-language (ancestor), Indo-European. English and German are part of a Germanic family branch, while French and Spanish belong to the Italic family branch of the Indo-European language family. See **Figure 2**.

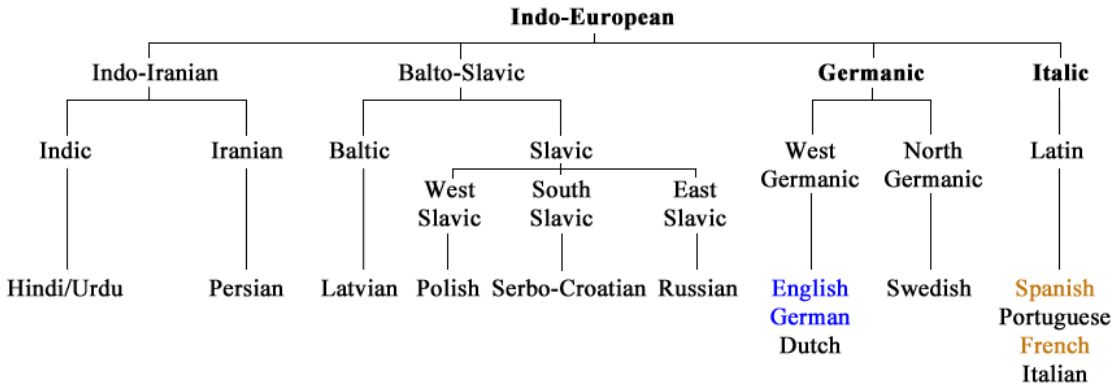


FIGURE 2 | A simplified Indo-European language family tree diagram adapted from Burkette, A., & Kretzschmar Jr., W. (2018). Historical Linguistics. In Exploring Linguistic Science: Language Use, Complexity, and Interaction (pp. 211-221).

TABLE 1 | Unique characters.

French	German	Spanish	English
é, è, à, ù, ç, â, ê, î, ô, û, ë, ï, ü, ÿ,œ,æ	ß, ö, ä	á, ñ, í, ó, ú	None

Being related these languages share a lot of identical or similar linguistic characteristics. French, German, Spanish, and English represent sounds using similar letters except for a few characters unique to these languages (**Table 1)**. These unique informative orthographic signs could be missed if stemming and lemmatization are applied during the preprocessing stage.

TABLE 2 | Examples of lexical similarity.

	French	German	Spanish	English
0	régulariser	regulieren	regularizar	regularize
1	activer	aktivieren	activar	activate
2	insectes	Insekten	insectos	insects

Lexical similarity is also observed among these related languages which makes it harder to distinguish between them. For example, the meaning of the words in **Table 2** could be understood by being able to speak just one of the languages. It is the morphology of the languages that helps with the identification. As shown in **Table 2**, the morphemes such as suffixes -er, -ieren, -ar, -ate in the word “activate” and endings -es, -en, -os, -s in the word “insects” help see that the words belong to different languages.

Among other common linguistic aspects of these languages is punctuation. The punctuation marks are very similar. Preserving punctuation to help distinguish between the languages makes sense if the punctuation marks are unique and not shared. 3 such unique signs: ¡ , ¿ (Spanish), and „ (German) are identified. They are preserved and used in the customized tf-idf.

The above-mentioned linguistic nuances are used to choose data preprocessing techniques and feature creation in this project.

DATA PREPROCESSING

The most commonly used words were identified for each of the four languages. As seen in **Figure 3** most of these words are stop words.

FIGURE 3 | Cloud words of most frequent words in 4 related languages

A few of these commonly used words are shared by the languages which does not help distinguish them. A decision was made to remove the most frequent shared words (see **Figure 4**) from the data. The unique stop words are kept and represent informative data.

FIGURE 4 | a list of shared most frequent words among the languages

```
['a', 'also', 'amazon', 'an', 'bien', 'color', 'da', 'de', 'dem', 'des',
'en', 'es', 'he', 'in', 'la', 'le', 'mal', 'material', 'me', 'mes', 'no',
 'on', 'que', 'se', 'si', 'so', 'son', 'super', 'top', 'un', 'was', 'y']
```

This project suggests preprocessing the variable “review_body” in two different ways:

- preprocessing typical of monolingual classification
- preprocessing that singles out and preserves as many unique linguistic features of each of the languages as possible

Table 3 summarizes the techniques used for each of the approaches:

TABLE 3 | Preprocessing techniques

Preprocessing *typical of monolingual classification*	Preprocessing while *preserving linguistic nuances*
1. Lowercase conversion	1. Lowercase conversion
2. Removing URLs and numbers	2. Removing URLs and numbers
3. Removing punctuation	3. Removing punctuation **except for ¡ , ¿ , and „**
4. Replacing contractions with a full form	4. Replacing contractions with a full form
5. Removing stop words for each of the languages	5. Removing **only** stop words **shared** by the languages
6. Stemming	6. **Skipping** stemming and lemmatization

Vectorizing the predictor variable "review_body" was done in different ways depending on the model type. For the statistical models, the data was vectorized using the TF-IDF typical of monolingual classification and the customized TF-IDF as seen in **Table 4** to see whether preserving linguistic characteristics affect the performance of the models.

TABLE 4 | Two ways of vectorizing preprocessed data for statistical models

Vectorization typical of monolingual classification	Vectorization while preserving linguistic nuances
1-3 word level n-grams in tf-idf	Customized tf-idf with 1-3 word level n-grams **preserving unique punctuation and stop words**

For the bidirectional LSTM, the data was vectorized using the embeddings. One of the biggest existing limitations of most of the multilingual embeddings is the fact that they are separate for each language and do not combine embeddings for several languages. FastText multilingual embeddings by Facebook are an exception and comprise embeddings for several languages having the same vector space, and accounting

for the similar meaning of a word in different languages. Yet, a lot of linguistic nuances are still not accounted for when using FastText multilingual embeddings. The decision was made to avoid using pre-trained multilingual embeddings and train embeddings as part of the neural network since the data set is very large and should be able to provide results comparable to high-quality pre-trained embeddings.

One-hot encoding was used on the target variable "language".

MODELING

Several types of models such as Multinomial Naïve Bayes, Logistic Regression, Random Forest, Support Vector Machines with the one-vs-one scheme, Support Vector Machines with the one-vs-rest scheme, and a Bidirectional LSTM were explored in this project and tested on long and short samples (see Appendix).

The statistical models named above were created using the predictor variable "review_body" preprocessed and vectorized accounting for the linguistic nuances of each of the languages and the one-hot encoded target variable "language". To avoid overfitting, cross-validation of the statistical models was performed. The resulting two statistical high-performing models (Multinomial Naïve Bayes and Logistic Regression) with the same parameters were also used with the data preprocessed and vectorized without accounting for linguistic nuances of each of the languages. These models demonstrated the lowest performance as seen in **Table 5.** This testifies to the fact that the preprocessing technique that keeps unique stop words for each of the languages, skips stemming, and preserves unique punctuation contributes to higher results in the identical models.

TABLE 5 | Table of models' accuracies

Model	Accuracy
Multinomial Naïve Bayes with tf-idf maximum features = 10,000	99.964%
Multinomial Naïve Bayes with tf-idf maximum features = 100,000	99.989%
Multinomial Naïve Bayes with tf-idf maximum features = 1,000,000	**99.994%**

Multinomial Naïve Bayes with tf-idf maximum features = 10,000 and preprocessing similar to monolingual classification	99.861%
Logistic Regression with with tf-idf maximum features = 10,000	99.961%
Logistic Regression with tf-idf maximum features = 10,000 and preprocessing similar to monolingual classification	99.849%
Random forest with tf-idf max features =10,000	99.925%
Support Vector Machines with tf-idf max features =10,000 and one-vs-rest scheme.	99.953%
Support Vector Machines with tf-idf max features =10,000 and one-vs-one scheme.	99.961%
Bidirectional LSTM	**99.99%**

Multinomial Naïve Bayes showed the highest accuracy results among the statistical models. Multinomial Naïve Bayes is a supervised algorithm relying on the Bayes theorem used when the number of classes is greater than 2.

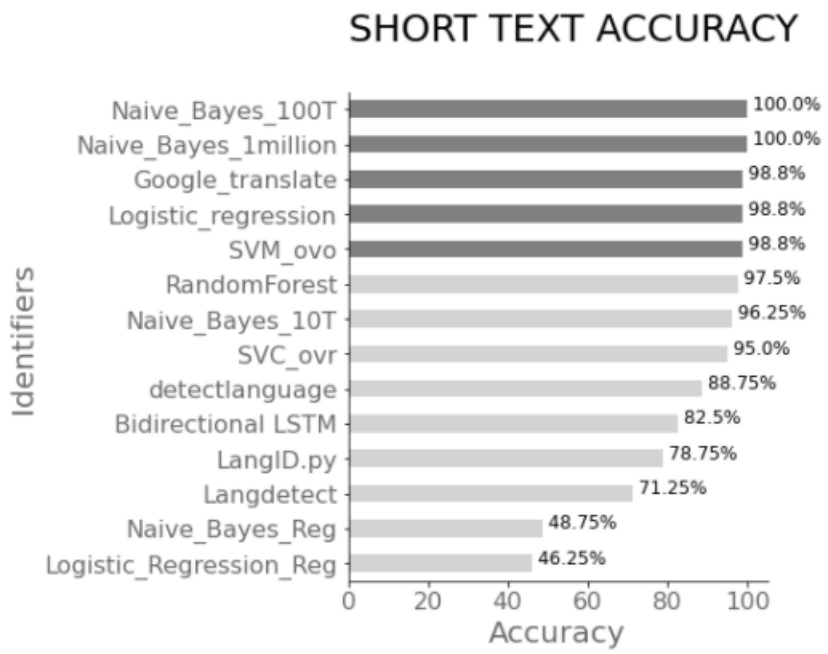
As seen in **Table 5** the highest accuracy of 99.994% is achieved by the Multinomial Naïve Bayes with the maximum features set to 1,000,000 and trained on the data preprocessed accounting for linguistic nuances of the languages. The results for the metrics of Recall, F1 and Precision match the accuracy results.

The bidirectional LSTM using the embedding layer and trained on the data preprocessed accounting for the linguistic nuances of the languages demonstrated high performance for the test and validation sets (more in Appendix). Its accuracy of 99.99 is comparable to that of the highest performing statistical model described above.

The performance of all of the models was further explored by testing them on short 3-word samples from the same data set but unseen by the models. Their results are also compared to the performance of the

existing language identification tools. The results in **Figure 4** show that the best performance (100% accuracy) was demonstrated by the Multinomial Naïve Bayes algorithms with the maximum features set to 100,000 and the Multinomial Naïve Bayes algorithms with the maximum features set to 1 million. Google translate API, Logistic Regression, and Support Vector Machines with the one-vs-one scheme demonstrated high accuracies as well. The bidirectional LSTM was not as accurate with 82.5% of the observations identified correctly.

FIGURE 4 | Comparison of accuracies in short text



Multinomial Naïve Bayes was previously used to create a language identifier Langid.py. The tool is tested in this project and identified 78.75% of the short strings correctly. Even though Langid.py and my approach use the same model type the preprocessing techniques and the vectorization of the data are different in this project. My choice of the word-level n-grams over byte-level n-grams in Langid.py, and a preprocessing of the data keeping the unique characteristics of the languages represent the differences between the approaches. The best performing models and tools were also tested on the formal text data (a sample from the European Parliamentary proceedings data set) to understand how well they perform

outside of the domain (**Figure 5**). Google Translate and Multinomial Naïve Bayes models demonstrated a perfect classification result while the bidirectional LSTM misclassified 12.5% of the observations.

FIGURE 5 | Accuracies on the data outside of the domain

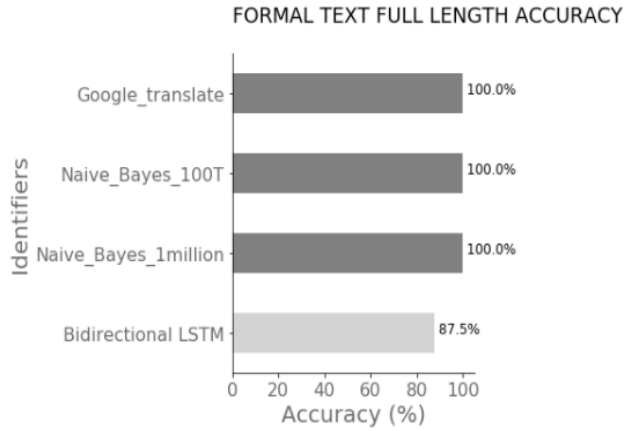
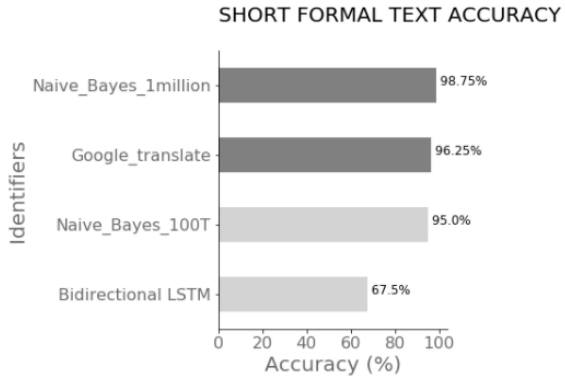


FIGURE 6 | Accuracies on the short text outside of the domain



The short version (3-word strings) of the formal text data samples was used to test how well the best-performing models and tools could identify the languages outside of the domain and on short text. **Figure 6** demonstrates the highest performance by the Multinomial Naïve Bayes with the maximum of features set to a million. Google Translate demonstrated a high accuracy but slightly lower than the best performing model created in this project. The lower performance of Google Translate could be explained by the fact that Google Translate supports more than 100 languages. A lot of these languages are related. When checking the languages that were misidentified (**Table 6)** by Google Translate API we can see that the misidentified results are languages from the same language family. The more related languages the models are trained on, the harder the task is.

TABLE 6 | Misidentified languages by Google Translate API

	Correct language	Misidentified language	Language family
1	German	Swedish	Germanic branch
2	Spanish	Galician	Romance branch
3	Spanish	Catalan	Romance branch

LIMITATIONS

The biggest limitation of the project is the number of languages used. The more languages there are, the higher the value of the tool is, and the better the capabilities of the tool are explored.

Another limitation is the domains. Even though both the formal and the informal texts were used to see how the models perform, only two samples were used. Testing on a larger number of samples from various sources can give a better idea of the language identifier's capabilities.

CONSIDERATIONS FOR THE FUTURE

The first step to improve the automatic language identifier created in this project is to expand the pool of languages. The bigger the number of languages used in the language identification tool the more linguistic effort needs to be involved to preprocess the data to preserve the most valuable information. The structure of different languages varies. For example, for the languages using hanzi/kana script such as Chinese or Japanese, the linguistic analysis would look different. Yet, I believe that even for these languages acknowledging and keeping the unique linguistic features when preprocessing the data can contribute to improvement in language identification.

CONCLUSION

In this project, two possible workflows for the multilingual analysis used to identify related languages from the Indo-European family have been studied. The performance of such models as Multinomial Naïve Bayes and Logistic Regression displayed that the data preprocessed relying on preserving linguistic characteristics of each of the languages contributed to the higher performance of the models than the data preprocessed similar to monolingual text analysis.

The importance of accounting for linguistic characteristics such as punctuation, unique stop words and morphemes when creating a language identifier is also apparent from the best models' performance on identifying a language of short texts as compared to the popular existing tools. 80 out of 80 short 3-word

balanced samples in German, Spanish, French, and English were identified accurately by the best performing Multinomial Naïve Bayes with the maximum features set to 1,000,000 that was trained on the data with preserved linguistic nuances. The bidirectional LSTM model though showing promising results for the long strings of test data, did not demonstrate comparable results for short samples.

The best performing models and tools were also tested on the data from the formal text domain to identify capabilities and possible weaknesses of the models. The results showed high accuracy for the Multinomial Naïve Bayes even on short 3-word text samples outside of the domain. The bidirectional LSTM model did not demonstrate comparable results outside of the domain.

To conclude, cross-validated Multinomial Naïve Bayes with the maximum features set to 1,000,000 and trained on the data with preserved linguistic nuances demonstrated the highest accuracy as compared to all of the models and tools tested in the project. The success of the model as compared to the previous research using Naïve Bayes for language identification lies mainly in the linguistic analysis involved and the preprocessing and vectorization techniques used in this project.

REFERENCES

1. Phillip Keung, Yichao Lu, György Szarvas and Noah A. Smith. “The Multilingual Amazon Reviews Corpus.” In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 2020.

2. Burkette, A., & Kretzschmar Jr., W. (2018). Historical Linguistics. In Exploring Linguistic Science: Language Use, Complexity, and Interaction (pp. 211-221). Cambridge: Cambridge University Press.

3. Jasanoff, Jay H., and Cowgill, Warren. "Indo-European languages". *Encyclopedia Britannica*, 13 Mar. 2020, https://www.britannica.com/topic/Indo-European-languages. Accessed 16 April 2021.

4. Pimental, I. 2020. The top 10 languages in higher demand for business. Retrieved from https://blog.amplexor.com/the-top-10-languages-in-higher-demand-for-business

5. Sutton, J. 2017. Introduction to Language Identification. Retrieved from https://algorithmia.com/blog/introduction-language-identification

6. Lui, Marco & Lau, Jey & Baldwin, Timothy. (2014). Automatic Detection and Language Identification of Multilingual Documents. Transactions of the Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/Q14-1003.pdf

7. Mathur, P., Misra, A., Budur, E. Language Identification from Text Documents. Retrieved from http://cs229.stanford.edu/proj2015/324_report.pdf

8. Gromann, D., Declerck, T., 2018. Comparing Pretrained Multilingual Word Embeddings on an Ontology Alignment Task. Retrieved from https://www.dfki.de/fileadmin/user_upload/import/9549_lrec-2018-3.pdf

9. Barba, P. 2020. Challenges Developing Multilingual Models in Natural Language Processing. Retrieved from https://towardsdatascience.com/challenges-in-developing-multilingual-language-models-in-natural-language-processing-nlp-f3b2bed64739

10. Gonzalez-Dominguez, J., Lopez-Moreno, I., Sak, H., González-Rodríguez, J., & Moreno, P. (2014). Automatic language identification using long short-term memory recurrent neural networks. Retrieved from https://www.isca-speech.org/archive/archive_papers/interspeech_2014/i14_2155.pdf

11. Lee, J. 2020. Benchmarking Language Detection for NLP. Retrieved from https://towardsdatascience.com/benchmarking-language-detection-for-nlp-8250ea8b67c

12. Lui, Marco & Lau, Jey & Baldwin, Timothy. (2014). Automatic Detection and Language Identification of Multilingual Documents. Transactions of the Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/Q14-1003.pdf

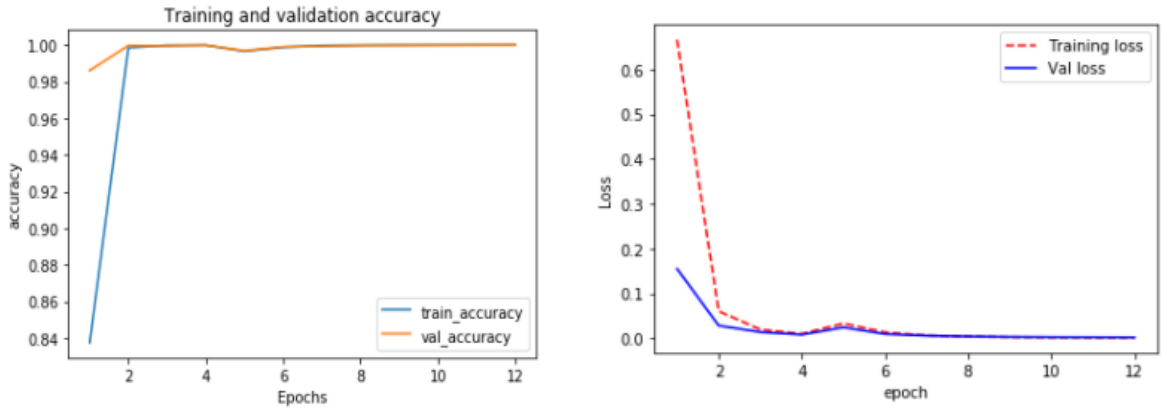
13. Koehn, Ph. Europarl: A Parallel Corpus for Statistical Machine Translation. Retrieved from https://www.statmt.org/europarl/archives.html#v6

Appendix

Models' results with preprocessing typical of monolingual classification	Models' results with preprocessing preserving linguistic nuances
Logistic Regression with tf-idf max features =10,000 accuracy: 99.849% (+/- 0.01%) precision: 99.849% (+/- 0.01%) recall: 99.849% (+/- 0.01%) f1 score: 99.849% (+/- 0.01%) Wall time: 10min 5s **Naïve Bayes** with tf-idf max features =10,000 accuracy: 99.861% (+/- 0.01%) precision: 99.861% (+/- 0.01%) recall: 99.861% (+/- 0.01%) f1 score: 99.861% (+/- 0.01%) Wall time: 7min 13s	**Logistic Regression** with tf-idf max features =10,000 accuracy: 99.961% (+/- 0.00%) precision: 99.961% (+/- 0.00%) recall: 99.961% (+/- 0.00%) f1 score: 99.961% (+/- 0.00%) Wall time: 14min 6s **Naïve Bayes** with tf-idf max features =10,000 accuracy: 99.964% (+/- 0.00%) precision: 99.964% (+/- 0.00%) recall: 99.964% (+/- 0.00%) f1 score: 99.964% (+/- 0.00%) Wall time: 10min 39s Naïve Bayes with tf-idf max features =**100,000** accuracy: 99.989% (+/- 0.00%) precision: 99.989% (+/- 0.00%) recall: 99.989% (+/- 0.00%) f1 score: 99.989% (+/- 0.00%) Wall time: 10min 59s Naïve Bayes with tf-idf max features =**1,000,000** accuracy: 99.994% (+/- 0.00%) precision: 99.994% (+/- 0.00%) recall: 99.994% (+/- 0.00%) f1 score: 99.994% (+/- 0.00%) Wall time: 11min 19s **Random forest** with tf-idf max features =10,000 accuracy: 99.925% (+/- 0.00%) precision: 99.925% (+/- 0.00%) recall: 99.925% (+/- 0.00%) f1 score: 99.925% (+/- 0.00%) Wall time: 31min 25s **Support Vector Machines** with tf-idf max features =10,000 and one-vs-rest scheme. accuracy: 99.953% (+/- 0.00%) precision: 99.953% (+/- 0.00%) recall: 99.953% (+/- 0.00%) f1 score: 99.953% (+/- 0.00%) Wall time: 12h 17min 32s **Support Vector Machines** with tf-idf max features =10,000 and one-vs-one scheme.

	accuracy: 99.961% (+/- 0.00%) precision: 99.961% (+/- 0.00%) recall: 99.961% (+/- 0.00%) f1 score: 99.961% (+/- 0.00%) Wall time: 4h 29min 25s

Bidirectional LSTM accuracy and loss details



We can see that the loss increased at epoch 5 but further training led to a steadily decreasing loss as seen in the training details below and the final accuracy as high as 99.99% at epoch 12:

```
Epoch 1/12
185/185 [==============================] - 2607s 14s/step - loss: 0.6660 - accuracy: 0.8375 - val_loss: 0.1550 - val_
accuracy: 0.9860
Epoch 2/12
185/185 [==============================] - 2400s 13s/step - loss: 0.0604 - accuracy: 0.9985 - val_loss: 0.0281 - val_
accuracy: 0.9996
Epoch 3/12
185/185 [==============================] - 2391s 13s/step - loss: 0.0194 - accuracy: 0.9997 - val_loss: 0.0137 - val_
accuracy: 0.9993
Epoch 4/12
185/185 [==============================] - 2385s 13s/step - loss: 0.0098 - accuracy: 0.9998 - val_loss: 0.0080 - val_
accuracy: 0.9998
Epoch 5/12
185/185 [==============================] - 2393s 13s/step - loss: 0.0328 - accuracy: 0.9965 - val_loss: 0.0242 - val_
accuracy: 0.9968
Epoch 6/12
185/185 [==============================] - 2394s 13s/step - loss: 0.0133 - accuracy: 0.9987 - val_loss: 0.0093 - val_
accuracy: 0.9989
Epoch 7/12
185/185 [==============================] - 2394s 13s/step - loss: 0.0062 - accuracy: 0.9996 - val_loss: 0.0058 - val_
accuracy: 0.9992
Epoch 8/12
185/185 [==============================] - 2397s 13s/step - loss: 0.0038 - accuracy: 0.9998 - val_loss: 0.0040 - val_
accuracy: 0.9995
Epoch 9/12
185/185 [==============================] - 2387s 13s/step - loss: 0.0022 - accuracy: 0.9999 - val_loss: 0.0029 - val_
accuracy: 0.9996
Epoch 10/12
185/185 [==============================] - 2390s 13s/step - loss: 0.0018 - accuracy: 0.9999 - val_loss: 0.0024 - val_
accuracy: 0.9997
Epoch 11/12
185/185 [==============================] - 2390s 13s/step - loss: 0.0014 - accuracy: 0.9999 - val_loss: 0.0022 - val_
accuracy: 0.9998
Epoch 12/12
185/185 [==============================] - 2398s 13s/step - loss: 0.0012 - accuracy: 0.9999 - val_loss: 0.0019 - val_
accuracy: 0.9999
```

Validation accuracy:

```
1282/1282 [==============================] - 277s 216ms/step - loss: 0.0022 - accuracy: 0.9998
```