Native-Language Identification with Attention

Stian Steinbakken and **Björn Gambäck**
Department of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway
stiansteinbakken94@gmail.com, gamback@ntnu.no

Abstract

The paper explores how an attention-based approach can increase performance on the task of native-language identification (NLI), i.e., to identify an author's first language given information expressed in a second language. Previously, Support Vector Machines have consistently outperformed deep learning-based methods on the TOEFL11 data set, the *de facto* standard for evaluating NLI systems.

The attention-based system BERT (Bidirectional Encoder Representations from Transformers) was first tested in isolation on the TOEFL11 data set, then used in a meta-classifier stack in combination with traditional techniques to produce an accuracy of 0.853. However, more labelled NLI data is now available, so BERT was also trained on the much larger Reddit-L2 data set, containing 50 times as many examples as previously used for English NLI, giving an accuracy of 0.902 on the Reddit-L2 in-domain test scenario, improving the state-of-the-art by 21.2 percentage points.

1 Introduction

Native-language identification (NLI) is the task of identifying an author's first language (L1; e.g., Spanish) given only information expressed in the author's second language (L2; e.g., English). NLI operates under the assumption that an author's L1 will dispose them towards particular language production patterns in their L2 (MacDonald, 2013). Knowing what mistakes a learner typically makes when writing or speaking in a foreign language can help educators create custom tutoring experiences and give feedback based on L1s. NLI also has applications within forensic linguistics, where identifying what country an unknown author is originally from is useful when detecting, e.g., plagiarism, web fraud and grooming, as well as in web-applications, web-crawling, and data collection to ensure high quality data.

Containing English learners of 11 different L1s, the TOEFL11 data set (Blanchard et al., 2013) has been the standard corpus for NLI. However, recent advances in auto-generating data based on social media users of high proficiency (Goldin et al., 2018) have enabled the collection of much larger data sets with more languages. State-of-the-art approaches on TOEFL11 reach over 0.88 accuracy in identifying the native language of the author using text only (Cimino and Dell'Orletta, 2017), while the best results on more English-proficient social media users approach 0.69 when evaluated on the same topics as trained on (Goldin et al., 2018). However, today's state-of-the-art systems are known for quite substantial drops in performance when tested on documents about topics not seen during training (Malmasi and Dras, 2018).

While the NLI field is evolving, so are advances in deep learning. Recently, attention-based models have shown promising results on various NLP tasks, and have become the *de facto* standard in sequence-to-sequence processing. Models such as the Transformer (Vaswani et al., 2017) rely solely on attention mechanisms, allowing for heavy parallelisation in the model training, as no recurrence or convolution is required. Using a network of transformers, BERT (Bidirectional Encoder Representations from Transformers) obtained new state-of-the-art results on 11 natural language processing tasks varying from question answering to sentence classification (Devlin et al., 2018).

Given the success of attention-based systems and the need for good NLI systems, the paper explores how such an approach can improve NLI performance, and investigates how robust attention-based systems are when tested on different topics than those they were trained on. Furthermore, while such systems perform well on their own, the paper also addresses how they can allow for improvements in combination with existing tech-

Proceedings of the 17th International Conference on Natural Language Processing, pages 261–271
Patna, India, December 18 - 21, 2020.

niques, given that all the best NLI systems to date include some ensemble or multi-classifier based architecture.

The paper is structured as follows: Section 2 describes the main data sets used in the field of NLI, while Section 3 covers related work. Section 4 introduces the model architectures, before Section 5 provides an overview of the experiments and their results. Finally, Section 6 sums up the findings and provides suggestions for further study.

2 Data Sets

The field of Native Language Identification is in a broader perspective quite young, and only gained serious momentum during the previous decade, most notably after Koppel et al. (2005) trained a Support Vector Machine (SVM) model on a vast amount of features including part-of-speech (POS) tags, n-grams and grammatical errors on ICLE, the International Corpus of Learner's English (Brooke and Hirst, 2013). Since then, two shared tasks have been dedicated to NLI, in 2013 and 2017. The experiments below and the discussed related work all focus on two data sets, TOEFL11 (the main data set used in the shared tasks) and Reddit-L2:

The **TOEFL11 data set**[1] (Blanchard et al., 2013) consists of English essays written by people with 11 different first languages for the college-entrance Test of English as a Foreign Language: Arabic, Chinese, French, German, Hindi, Italian, Japanese, Korean, Spanish, Telugu and Turkish. The corpus contains 13,100 essays, with 1,100 essays per language distributed as evenly as possible across 8 different topics. Each essay is also labelled with its score (low, medium or high). For the 2013 shared task, the data set was split into a training set consisting of 900 essays per L1, and development and test sets consisting of 100 essays each per L1.

The **Reddit-L2 data set**[2] was collected by Rabinovich et al. (2018) from the social media platform Reddit, where users can subscribe to areas of interest, called "subreddits" (Singer et al., 2014). These subreddits vary from general topics such as "pictures" to specific niche areas such as "birds with arms" (which is for posting pictures of birds photoshopped to look like they have arms). Some subreddits are dedicated to Europe, where the users can optionally provide a tag ('flair') indicating their home country. These flairs were utilised by Rabinovich et al. to infer the users' L1, with experiments performed to quantify the reliability of the labels. Goldin et al. (2018) added further measures to verify the labels' correctness.

While TOEFL11 contains texts written by learners of English, many of the Reddit-L2 authors are close to fluent in English, making the task of performing NLI on this data set more challenging. The data set consists of texts the users have produced in Europe-related subreddits (*in-domain*), as well as texts written in other subreddits, that can have virtually any topic (*out-of-domain*). The data is separated into chunks of 100 sentences from one user. After pre-processing and removing users with less than 100 sentences, the Reddit-L2 data set consists of roughly 200 million sentences and 3 billion tokens across 29 native countries, from 34,511 unique users.

The number of users varies between countries, making the data set imbalanced. Hence Goldin et al. (2018) downsampled the data, grouping 29 native countries into 23 languages: English, German, Dutch, French, Polish, Romanian, Finish, Swedish, Spanish, Greek, Portuguese, Estonian, Czech, Italian, Russian, Turkish, Bulgarian, Croatian, Norwegian, Hungarian, Lithuanian, Slovenian and Serbian. The number of users per label was capped at 104, which is the number of users present for the labels with fewest users (Lithuania and Slovenia). For labels with more than 104 users available, 104 users were selected at random. For each user, the median number of chunks was selected. 3 chunks per user for an in-domain scenario, and 17 chunks per user for an out-of-domain scenario,

3 Related Work

As mentioned in the previous section, two shared tasks have been dedicated to NLI, taking place in 2013 and 2017, and with respectively 29 and 19 teams participating. The **2013 NLI Shared Task** (Tetreault et al., 2013) introduced the TOEFL11 corpus and was divided into three sub-tasks: Closed-Training, Open-Training-1 and Open-Training-2. The first was the main task, allowing usage of the training and development sets only. The Open-Training-1 task allowed the use of any training data *except* TOEFL11, while Open-Training-2 allowed the use of any data, including TOEFL11. However, Open-Training-1 showed that training on external corpora while testing on TOEFL11 caused a significant drop in accuracy.

[1] catalog.ldc.upenn.edu/LDC2014T06
[2] cl.haifa.ac.il/reports/L2/index.shtml

The most common features used were word, character and part-of-speech n-grams. Four of the top five teams used at least word 4-grams, and some as high as 7- and 9-grams. The best team achieved an accuracy of 0.846 (Tetreault et al., 2013), and like the overwhelming majority of the participating teams used SVMs. Tetreault et al. (2012) additionally improved performance using ensemble methods. Later, Ionescu et al. (2014) improved the results further using string kernels.

The **2017 NLI shared task** (Malmasi et al., 2017) was similar to the 2013 task, but in addition to the text utilised TOEFL11 data from a speech-based NLI task included in the 2016 INTERSPEECH Computational Paralinguistics Challenge (Schuller et al., 2016). The raw speech data could not be distributed in that challenge, so the data set consisted of textual transcripts together with so called i-vectors, which are vectors of fixed length (here 800), working as lower-dimensional representations of high-dimensional sequential speech recordings.

Hence the 2017 shared task was divided into three tracks: essays only, speech only, and a combined 'fusion' track using both text and speech. The fusion track submissions showed that combining written and spoken responses provided a large boost in prediction accuracy (Malmasi et al., 2017). Furthermore, ensemble-based systems were the most effective in all tasks, but typically the same features were used as in 2013, and SVMs were still the most popular approach, dominating deep learning models. Ircing et al. (2017) hypothesise that this could be due to the size of the TOEFL11 data set, and that more training examples could help deep learning models perform better.

UnibicKernel (Ionescu and Popescu, 2017) performed best on the fusion track (0.932 accuracy),[3] topped the speech-only track, and placed in the top tier in the essay track, by using multiple kernel learning and kernel discriminant analysis, KDA. KDA is a kernelised version of LDA, Linear Discriminant Analysis, that is, a method for finding the best linear combination of features that characterise or separate two or more classes, by projecting the data points down from a feature space where they not are linearly separable to a lower dimensional space where they are.

[3]The 2017 shared task used macro-averaged F_1 score as the official evaluation metric, but also reported accuracy. However, accuracy will be used here for consistency, and since the F_1 scores were typically almost identical to the accuracy.

Also finishing in the top group was CEMI (Ircing et al., 2017), using a neural network based meta-classifier of several isolated feed-forward neural network (FFNN) models, each trained on a separate feature type (such as word, character and POS n-grams, plus i-vectors), with the outputs combined using softmax to predict the final label.

The ItaliaNLP team (Cimino and Dell'Orletta, 2017) topped the essay-only track with 0.882 accuracy, which currently is the highest score achieved on the textual TOEFL11 test set. They used two stacked SVM-classifiers, one trained at sentence level and the other at document level, using the output of the sentence classifier as input, together with features from the documents themselves: text and average word length; function words; as well as character, word, lemma, POS, and linear dependency n-grams.

After the shared tasks, Malmasi and Dras (2018) performed a systematic examination of ensemble methods for NLI, in addition to evaluating deeper ensemble architectures such as classifier stacks. The experiments included a rigorous application of meta-classification models, achieving state-of-the-art results on several large data sets, evaluated both intra-corpus and cross-corpus. Two important trends were observed: the meta-classification results were better than the ensemble combination methods alone, and meta-classifiers trained on continuous output performed better than their discrete label counterparts. The best performing meta-classifier was LDA, both individually and as an ensemble. The ensemble of LDA meta-classifiers obtained an accuracy of 0.871 — close to the state-of-the-art accuracy of 0.882 obtained by ItaliaNLP.

Goldin et al. (2018) tried the NLI task in a new environment using the larger **Reddit-L2** data set. A simple regression classifier was trained for all experiments, since the main focus was not to build a new NLI classifier, but to explore possible features specific to social media and the Reddit-L2 data. They used content features (3-grams and POS 1-grams), spelling and grammar features, and content-independent features such as function words, the most frequent POS 3-grams in the data set, and average sentence length. Furthermore, Goldin et al. experimented with social network specific features such as votes from other users, average number of submissions and comments, and the most frequent subreddits the user had visited, compared to the most popular subreddits for each country.

The most visited subreddits feature performed stunningly, both in and out of the domain. However, this feature is not only specific to the particular data set, but as many users tended to frequent subreddits specific to their country, the feature often contained the correct label itself. Hence Goldin et al. reported the results of using all features, excluding the Subreddit feature, which yielded an accuracy of 0.690 in the in-domain scenario, dropping down to 0.36 when tested out-of-domain.

4 Architecture and Model

Three model architectures were used in the experiments below: a stand-alone BERT model tailored to NLI, a meta-classifier architecture, and an ensemble of meta-classifiers. The stacked classification architectures use traditional techniques in combination with BERT. The PyTorch version of BERT was used,[4] which is a clone of Google's official BERT implementation for TensorFlow,[5] using the same pre-trained models as provided by Devlin et al. (2018). All other classifiers were implemented using the open source Scikit-Learn (Sklearn) Machine Learning library.[6] For SVMs the default parameters of Sklearn were used, but with a linear kernel, as it was the kernel used for all base classifiers in Malmasi and Dras (2018). The FFNNs (feed-forward neural networks) were run with the default parameters of Sklearn, unless specified otherwise. Sklearn was also used for feature manipulation (e.g., transforming text into TF-IDF weighted vectors), with Natural Language Toolkit (NLTK)[7] used for basic text-processing, and Pandas[8] for data manipulation.

All experiments were carried out on a two GPU cluster, with 16 GB each. For most jobs, 64 GB RAM was sufficient, but for the largest job 128 GB RAM was needed in order to keep both BERT and all the data in memory. Running BERT on TOEFL11 for 5 epochs typically took 2–3 hours, while experiments using millions of Reddit-L2 examples took more than 6 days. All the code from the paper is available on GitHub.[9]

[4] github.com/huggingface/pytorch-pretrained-BERT
[5] github.com/google-research/bert
[6] scikit-learn.org/stable/
[7] www.nltk.org/
[8] pandas.pydata.org/
[9] github.com/stianste/BERT-NLI

4.1 BERT Model Architecture

The overall BERT architecture consists of the pre-trained BERT model followed by a linear layer, as suggested by Devlin et al. (2018). The linear layer is randomly initialised and must be trained from scratch for each classification task. It has an input size the same as the BERT hidden size output and an output size matching the number of labels for the relevant task (11 for TOEFL11 and 23 for Reddit-L2). The BERT model and the linear layer are trained together using the cross-entropy loss with regards to the correct label.

To reduce vocabulary size and tokenisation complexity, the uncased version of BERT was used in all experiments, although including capitalisation could in some cases be beneficial for NLI. Following Devlin et al. (2018), all experiments utilised the same hyper-parameter settings, except for three that were varied in the ranges found by them: batch size $\{16, 32\}$, Adam learning rate $\{5e^{-5}, 3e^{-5}, 2e^{-5}\}$, and number of epochs $\{3, 4\}$, with fixed numbers of hidden layers (12), hidden size (768), attention heads per transformer (12), proportion of training to perform linear learning rate warmup for (0.1), and gradient accumulation steps (1).

However, while Devlin et al. restricted the maximum total input sequence length after WordPiece tokenisation to 128, it was here set to its largest possible value, 512, since most documents available in the NLI data sets contain far more tokens than 128: the average and maximum number of WordPiece tokens per document is 369 and 910 for TOEFL11, growing to 2,072 and 18,149 for Reddit-L2. 98.9% of the TOEFL11 documents contain more than 128 tokens, but only 4.9% have more than 512 tokens.

In the Reddit-L2 data set, however, all documents surpass 512 tokens, so it must be split up into chunks, in order not to waste data. A heuristic division of the documents was applied, dividing all examples into smaller sub-examples (sub-chunks) with a length of roughly 512 tokens, by splitting on spaces. After training has been performed on the sub-chunks, evaluation was carried out both on the individual sub-chunks and on the recombination back into the original chunks, using a majority vote based on each sub-chunk prediction. Ties were broken randomly.

It is important to note that the previous work on Reddit-L2 by Goldin et al. (2018) evaluated on the prediction of each chunk. For this reason, the accuracies obtained at the chunk level will be the

comparable measurement of the system's performance, not the *ad hoc* sub-chunks that have only been created in order to fit the limitations of BERT.

Preliminary experiments further showed that BERT-large with a maximum sequence length of 512 caused memory issues with the GPUs available, as did a batch size of 32 for BERT-base. The largest models that were able to run with a 512 sequence length were BERT-base with a batch size of 16 or less, and BERT-large with a batch size of 1. Hence all experiments used the BERT-base model, except when running BERT-large with batch size 1. This is unfortunate, as Devlin et al. (2018) report that BERT-large provides a significant performance boost over BERT-base, even for small data sets.

4.2 Meta-Classifier Architectures

To explore how an attention-based system can be used in union with the current-state-of-the art approaches to improve NLI performance, two novel architectures were tested, inspired by the classifier-stacking of Malmasi and Dras (2018), but including BERT as an optional base-classifier.

The first is a stacked architecture consisting of homogeneous base classifiers, with their combined output fed into a meta-classifier providing the final decision. An illustration of the meta-classifier architecture can be found in Figure 1. The inclusion of BERT as a base-classifier is optional, indicated by stippled lines. All base classifiers are fed the raw text input, but trained on different feature types.

After training, each base classifier produces a vector of continuous probabilities for each class. This is used since Malmasi and Dras (2018) showed continuous probability output in general yielding better performance than discrete one-hot encoded label representation. The meta-classifier is then trained on the concatenated output of the base classifiers and the original training labels, to produce a single prediction per example instance.

The second architecture is similar to the first, but instead of using a single meta-classifier, an ensemble of meta-classifiers is applied to provide the final decision of the stack. It follows the best performing ensemble found by Malmasi and Dras (2018), with 200 meta-classifiers merged in a bagging-ensemble, where all models are trained on different subsets of the base classifiers.

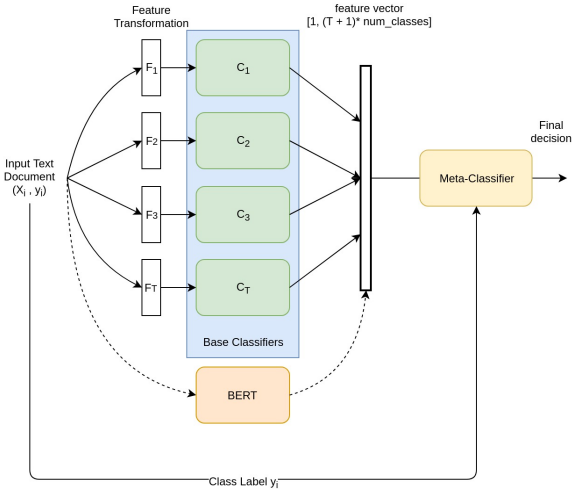

Figure 1: A meta-classifier stack with BERT as an optional base classifier

5 Experiments and Results

To set reasonable baselines for the experiments, a simple multinomial Naïve Bayes (MNB) and an SVM classifier were trained and evaluated on the unigram representation of the documents. As done by Malmasi and Dras (2018), the unigram feature vector was normalised using each word's TF-IDF score. Initial experiments were then run to give an indication of how an attention-based architecture in isolation performs on the NLI task, as well as when tested on documents on topics different from what it was initially trained on. Further experiments addressed whether the combination of an attention-based system and the techniques used in the current state-of-the-art can improve performance. Finally, it was investigated how the attention-system performs with more data available.

5.1 Attention-based System Only

The first set of experiments aimed to explore how an attention-based system alone performs on the NLI task. These initial experiments also indicated what hyper-parameters the BERT model performs best under. As most of the related literature trains on the TOEFL11 training set and evaluates on the official TOEFL11 test set, that setup was applied. Similarly, experiments on Reddit-L2 in-domain data set applied the same downsampling and 10-fold cross-validation as used by Goldin et al. (2018). To test how an attention-based system performs on the task of NLI when tested on documents concerning topics different from what it was initially trained on, the out-of-domain experimental setup

MNB	SVM	BERT			
		$2e^{-5}$	$3e^{-5}$	$4e^{-5}$	$5e^{-5}$
0.559	0.726	0.759	**0.777**	0.761	0.765

Table 1: TOEFL11 accuracies, different learning rates

described in Goldin et al. (2018) was followed, i.e., training the model on the in-domain data, and testing on out-of-domain data of different users.

The results of the experiments on **TOEFL11** are found in Table 1. As expected, the SVM classifier clearly outperforms Naïve Bayes. The best results using BERT were obtained with a learning rate of $3e^{-5}$, with a final accuracy of **0.777**. Under these hyper-parameters, the model obtained a training loss of 0.110 and an evaluation loss of 0.824. Experiments were run over 3, 4, 5 and 10 epochs, but the table only reports the results after 5 epochs, since increasing epochs typically increased accuracy, regardless of the learning rate. However, the 10-epoch model achieved an accuracy slightly below the best at 5 epochs, with a training loss of 0.004, but an evaluation loss of 1.140 on the test set, indicating that it overfitted the training data.

The BERT experiments reported in Table 1 were run with a constant batch size of 16. Testing different batch sizes $\{1, 2, 3, 4, 8\}$ under the optimal settings (a learning rate of $3e^{-5}$ over 5 epochs) gave no indication that a higher or lower batch size is better or worse, so the remaining experiments were run with size 16 batches, in order to keep the training time as low as possible without encountering memory issues or reducing model performance.

For comparison, an experiment running BERT-large with a batch size of 1 was also carried out, using the best performing learning-rate and number of epochs found for BERT-base. However, BERT-large was not able to converge under these settings: after 5 epochs it produced the same probability for each L1 for all test cases. Hence BERT-large was run with both a smaller and larger learning rate. While larger learning rates did not improve matters, a learning-rate of $2e^{-5}$ produced a final accuracy on the TOEFL11 test set of 0.759. BERT-base with a batch size of 1 obtained an accuracy of 0.770, so BERT-large does not seem to provide any significant benefit over BERT-base for TOEFL11.

Table 2 shows average accuracy over 10-fold cross-validation on the **Reddit-L2** data set. After downsampling the data set and splitting all documents into sub-documents of max 512 tokens,

Scenario	MNB	SVM	BERT
In-domain, chunks	0.377	0.716	**0.805**
In-domain, sub-chunks	0.350	0.574	0.651
Out-of-domain, chunks	0.176	0.400	**0.502**
Out-of-domain, sub-chunks	0.169	0.322	0.400

Table 2: Reddit-L2 accuracies

there were roughly 18,300 training examples and 1,943 test cases per fold. Both sub-chunk and chunk/document accuracy are reported. After recombining all sub-chunks, there was an average of 500 test chunks per fold.

As expected based on related work, the simple unigram SVM baseline improves on the 0.690 accuracy Goldin et al. (2018) obtained on the in-domain scenario using logistic regression. The unigram SVM with an accuracy of 0.400 also beats the previous best out-of-domain score of 0.362. The accuracy of both Naïve Bayes and SVM drop when tested in the out-of-domain Reddit-L2 scenario, and further drop when the models are evaluated on the more granular sub-chunks. The final average in-domain accuracy of BERT across the ten folds was **0.805**, a substantial improvement over the SVM classifier and over the current state-of-the-art (0.690). When evaluating on each individual sub-chunk, the accuracy drops to 0.651, indicating that the model predicts parts of the chunks incorrectly, but performs well on the documents/chunks as a whole using majority vote.

Running BERT in the out-of-domain Reddit-L2 scenario, the final average over random 10-fold cross-validation was **0.502**, a clear improvement over the single SVM baseline of 0.400, and again a substantial improvement over the 0.362 obtained by Goldin et al. (2018). Note that the train-test split in the out-of-domain scenario can be regarded as unfavourable: After downsampling, the 10% out-of-domain test chunks outnumber the 90% in-domain training chunks, with most folds having 16,000–16,500 training examples, but 20,000–21,000 test cases. Commonly, training sets are 5–10 times as big as test sets, but with the out-of-domain scenario the training set is smaller than the test set, making for a rigorous evaluation scenario.

Comparing the out-of-domain results to the in-domain results, BERT's accuracy drops from 0.805 to 0.502, a 37.9% relative drop in accuracy, compared to the 47.5% drop obtained by Goldin et al. (2018), indicating that BERT is more robust than logistic regression when tested off-topic.

Base Classifier / Meta-Classifier	SVM	FFNN
SVM	0.756	0.745
SVM + BERT	0.794	0.791
FFNN	0.819	0.825
FFNN + BERT	0.838	**0.853**

Table 3: Meta-classifier results on TOEFL11

Base Classifier / Ensemble	SVM	FFNN
SVM	0.755	0.801
SVM + BERT	0.798	0.823
FFNN	0.827	0.808
FFNN + BERT	0.849	**0.851**

Table 4: Ensemble of meta-classifiers on TOEFL11

5.2 Combining Attention with State-of-the-Art Techniques

The next set of experiments utilised the meta-classifier and ensemble architectures described in Section 4.2. A grid search over several hyper-parameters was carried out, over the different base classifiers, the types of features per classifier, and the maximum number of features per classifier. The different models and feature types were first explored individually, in order to assess their individual contribution to the meta-classifier or ensemble. Both SVMs and FFNNs were used as base classifiers. Based on the features used by Ircing et al. (2017) and Malmasi and Dras (2018), word 1–3 grams, character 1–4 grams, and lemma 1–2 grams were used, in addition to content-independent function word 1–2 grams. Similarly to Malmasi and Dras (2018), the main focus was on evaluating whether the inclusion of BERT can improve the results obtained by using a predefined set of features, rather than on feature exploration.

The SVM and FFNN base classifiers were run with the TF-IDF weighted representation of each feature-type, with the maximum number of features per example capped at 5,000, 10,000, 30,000, and no limit. Word 3-grams performed surprisingly bad compared to word 1- and 2-grams, and the content-independent function word features performed far worse than the content-dependent features. The single best performing feature type using both SVM and FFNN was character 4-grams. The SVM model favoured feature vectors of size 30,000, while the FFNN performed better with no feature limit, potentially since the FFNN is better at filtering out the noise obtained by including more TF-IDF features, while still finding useful information in more than 30,000 features.

The results of training a **meta-classifier** on the continuous probability outputs of the ten base classifiers from the previous experiment are found in Table 3. The inclusion of BERT means appending the $n\_classes$ logit outputs of BERT to the training and test data. Experiments were also carried out normalising BERT's output using softmax and raw probabilities, but using the raw logit output performed better overall.

The inclusion of BERT has a significant impact on the performance: using SVMs as base and meta-classifier, the accuracy increases from 0.756 to 0.794. The FFNN trained on the SVM base classifiers reap even more benefits when including BERT, increasing from 0.745 to 0.791. Perhaps surprisingly, FFNNs perform best both as base-classifier and meta-classifier: the accuracy obtained by using FFNN base classifiers and BERT with a FFNN meta-classifier is **0.853** (up from 0.825), slightly below the current best text-only score on TOEFL11 test, 0.882 by Cimino and Dell'Orletta (2017).

The results of training an **ensemble of meta-classifiers** can be found in Table 4. Again, the inclusion of BERT causes a significant 4.3 percentage point increase for the best performing ensemble. However, contrary to the results of Malmasi and Dras (2018), there was no clear gain in performance when applying a single meta-classifier as opposed to using a bagging ensemble of meta-classifiers. This is surprising and interesting, as the experiments and setup are quite similar.

Looking more closely at the results of Malmasi and Dras, though, the increase in accuracy when going from a single meta-classifier to an ensemble of meta-classifiers was only present for the LDA meta-classifier, while the other meta-ensembles (e.g., the SVM-based one) obtained the exact same TOEFL11 test set accuracy as their corresponding meta-classifiers. Thus, the lack of increase in performance when using an ensemble of meta-classifiers as opposed to using a single one could actually be expected for the SVM and FFNNs used.

Experiments using LDA as the meta-classifier were also carried out, but provided disappointing results, a lot lower than the best single base classifier, potentially due to the collinearity of the features used. Using a more diverse feature set might have made LDA a more viable candidate, both as a single meta-classifier and in an ensemble.

Classifier / Ensemble	In-domain	Out-of-domain
FFNN meta-classifier	0.765	0.452
FFNN + BERT	**0.818**	**0.529**
BERT alone	0.805	0.502

Table 5: Ensemble of meta-classifiers on Reddit-L2

The stack setup which performed best on TOEFL11 (i.e., the FFNN meta-classifier trained on the outputs of the base FFNN classifiers and BERT) was run also on the **Reddit-L2** in- and out-of-domain scenarios (Table 5). Using the same 10 folds as the previous in-domain experiment, the ensemble *without* BERT obtained an accuracy of 0.765. The same ensemble *with* BERT boosted the accuracy to **0.818**, Hence BERT seems to be doing most of the heavy lifting in the ensemble in the in-domain scenario. However, the ensemble also provides BERT with a minor performance boost, increasing accuracy by 1.3 percentage points compared to using BERT alone on the task.

The final average accuracy over the 10 out-of-domain folds obtained by the meta-classifier was 0.452 without using BERT. Including BERT in the ensemble yielded an average accuracy of **0.529**. This seems comparable to the in-domain results, as the final meta-classifier accuracy again is slightly higher than that obtained by using BERT alone.

5.3 More Data

While the Reddit-L2 data set is huge compared to previous NLI data sets, most of the data is discarded when performing downsampling, which is done to maintain class balance. Using evaluation metrics that take class imbalance into consideration, such as the macro-averaged F_1 score, can mitigate this problem. Hence BERT was trained on the out-of-domain Reddit-L2 data, and then the entire in-domain data set was used for testing, to show how the same attention-system as used alone in the first experiments performs with more data available. Training on only out-of-domain data will also display how the system reacts when trained on some topics and tested on others.

To maintain a fair training-to-test ratio, the out-of-domain training data was engineered to be roughly 10 times the size of the in-domain test set, and was balanced to contain roughly the same number of examples per label. This was done by capping the maximum number of training sub-chunks per label to 80,000, leaving 1,491,198 sub-chunks for training, with 282,385 for testing. This heuristic division causes some class imbalance in the training data, as some languages have less than 80,000 sub-chunks, with Slovenian only having 24,787 sub-chunks available out-of-domain. This issue will have to be tolerated, however, as the intent of the experiment was to explore how more data impacts performance. Using macro-averaged F_1 as evaluation metric will also account for this bias to some extent, by measuring performance with the same weighting for each class, An alternative solution would be to set the cap at 24,787 sub-chunks per class, leaving only 570,101 training instances.

After recombining the sub-chunks into the original chunks, there were a total of 71,716 test chunks, far more than the roughly 500 test chunks per fold in the in-domain scenario of Goldin et al. (2018) and the previous experiments. The final accuracy obtained by BERT on these test chunks in the in-domain Reddit-L2 data set was **0.861**. For comparison, the Naïve-Bayes unigram baseline classifier achieved an accuracy of **0.278** when trained and tested on the same data, while the SVM classifier failed to finishing training. As the in-domain test set is not fully balanced with regards to classes, with English having 27.8% of the test labels, the accuracy achieved might have been artificially high. However, the final macro-average F_1 score obtained was **0.847**, indicating that the model is performing well over all classes.

Inspecting the confusion matrix of the predicted outputs, the model seemed to be slightly biased towards predicting English, in particular when the correct label was French or Dutch. Interestingly, the model confused Norwegian with Swedish, and vice versa, both of which were also mistaken for English in 4% of the cases. As for the classes with the fewest training and test instances, such as Serbian, Slovenian and Lithuanian, the model seemed to have no problem, as it obtained 0.86, 0.85 and 0.91 accuracy internally within those classes. In fact, both Turkish and Estonian have rather few training examples, and less than 1,600 test cases, but the model achieved 0.98 and 0.94 in classifying these labels, respectively.

When evaluating this model on the same 10 folds as used in the in-domain scenario in the first experiment, it obtained a final accuracy of **0.902**, and an F_1 score of **0.901**. This task can be considered easier than the task of predicting the in-domain data set as a whole, as the average number of chunks per fold were roughly 500, as opposed to the 70K

test cases when testing on the entire in-domain set. However, the 10 downsampled folds are balanced for classes, making the task non-trivial. The accuracy of 0.902 indicates that BERT thrives with more data available, even though the out-of-domain examples used for training contain no specific topic.

6 Conclusion

The paper is the first to apply BERT to the native language identification task. An empirical exploration showed that BERT alone is not able to compete with traditional state-of-the-art approaches on the TOEFL11 data. However, a meta-classifier architecture combining BERT and ten traditional classifiers trained on different features produced an accuracy of **0.853** on the TOEFL11 test set, closing in on the current state-of-the-art of 0.882.

Further experiments showed that BERT can produce state-of-the-art results on the novel Reddit-L2 data set, both in- and out-of-domain, with a **0.902** 10-fold cross-validated accuracy for the in-domain scenario, with a model trained only on out-of-domain data, but tested on in-domain data. This can be compared to the result of 0.690, obtained by Goldin et al. (2018) when both training and testing the model on in-domain data. With the same setup, BERT achieved an accuracy of 0.805.

Additionally, when evaluated on the entire in-domain test set as a whole (70,000 test cases versus roughly 500 test cases for each fold in the previous in-domain scenario), BERT was able to obtain an even higher accuracy of 0.861. This is not directly comparable to any related work, as the entire in-domain part of the data set has not before been used for testing; however, this task is potentially more demanding than the original in-domain task, as it contains 140 times as many test cases in each fold.

Testing the BERT model out-of-domain showed that it is more robust to topic differences than previous approaches on Reddit-L2, both BERT alone and in a meta-classifier stack, which produced a final accuracy of **0.529**, a 16.7 point improvement over the state-of-the-art.

The performance of BERT with more data available indicates that BERT's performance increases with more data, regardless of what topics it is trained on. In addition to the document granularity the model is trained on, a further reason for why BERT performs so much better on Reddit-L2 than on TOEFL11 can be the number of spelling mistakes found in the TOEFL11 data. This is due both to the higher proficiency of the Reddit-L2 users and to them having spellcheckers and other tools available. A higher frequency of misspelled words can cause the WordPiece representations used by BERT to be different from those which it was pre-trained on. This might be mitigated by further training of the pre-trained models from BERT's check points on data containing spelling mistakes, providing a custom BERT model with embeddings tuned to include spelling mistakes.

As BERT obtained such promising results on the Reddit-L2 in-domain data set, future work should look into how to increase the accuracy on TOEFL11, the standard data set for NLI. In addition of further pre-training of BERT to learn embeddings of spelling errors and other domain attributes, this would include using additional information sources, such as the sentiment and emotions reflected in the texts, as done by Markov et al. (2018b), or including information about punctuation (Markov et al., 2018a) or capitalisation. Clearly, training on more relevant data would also improve performance, so including training data from the italki corpus (Hudson and Jaf, 2018) should be feasible. It contains about 122,000 documents gathered from the language learning site italki, in the same languages as TOEFL11.

With more power available, running BERT-large with a sequence length of 512 should be feasible, or alternatively extensions of BERT or similar models, such as ALBERT (A lite BERT; Lan et al., 2020), GPT-3 (Generative Pre-trained Transformer; Brown et al., 2020) continuous pre-training ('ERNIE 2.0'; Sun et al., 2020) or transformers for longer sequences ('BigBird'; Zaheer et al., 2020) could be tested on the problem.

Opening up the attention mechanism and looking at what parts of the input sequence the model is paying attention to could also give new insights, and could potentially help educators.

Acknowledgments

Thanks to Hans Olav Slotte, Iselin Eriksen, Charles Edvardsen and Vebjørn Isaksen for good discussions on NLI and BERT.

Further thanks to Rabinovich et al. for making the Reddit-L2 data set openly available online, to Devlin et al. for distributing the pre-trained BERT models and their code open source, and to Malmasi and Dras for answering questions about their LDA classifier.

References

Daniel Blanchard, Joel Tetreault, Derrick Higgins, Aoife Cahill, and Martin Chodorow. 2013. TOEFL11: A corpus of non-native English. *ETS Research Report Series*, 2013(2):i–15.

Julian Brooke and Graeme Hirst. 2013. Using other learner corpora in the 2013 NLI shared task. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 188–196, Atlanta, Georgia. Association for Computational Linguistics.

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. *CoRR*, abs/2005.14165.

Andrea Cimino and Felice Dell'Orletta. 2017. Stacked sentence-document classifier approach for improving native language identification. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 430–437, Copenhagen, Denmark. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Gili Goldin, Ella Rabinovich, and Shuly Wintner. 2018. Native language identification with user generated content. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3591–3601, Brussels, Belgium. Association for Computational Linguistics.

Thomas G. Hudson and Sardar Jaf. 2018. On the development of a large scale corpus for native language identification. In *Proceedings of the 17th International Workshop on Treebanks and Linguistic Theories*, pages 115–129, Oslo, Norway. Linköping University Electronic Press.

Radu Tudor Ionescu and Marius Popescu. 2017. Can string kernels pass the test of time in native language identification? In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 224–234, Copenhagen, Denmark. Association for Computational Linguistics.

Radu Tudor Ionescu, Marius Popescu, and Aoife Cahill. 2014. Can characters reveal your native language? A language-independent approach to native language identification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1363–1373, Doha, Qatar. Association for Computational Linguistics.

Pavel Ircing, Jan Švec, Zbyněk Zajíc, Barbora Hladká, and Martin Holub. 2017. Combining textual and speech features in the NLI task using state-of-the-art machine learning techniques. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 198–209, Copenhagen, Denmark. Association for Computational Linguistics.

Moshe Koppel, Jonathan Schler, and Kfir Zigdon. 2005. Determining an author's native language by mining a text for errors. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 624–628, Chicago, Illinois, USA. ACM.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *Eighth International Conference on Learning Representations*, Addis Ababa, Ethiopia. ICLR.

Maryellen C. MacDonald. 2013. How language production shapes language form and comprehension. *Frontiers in Psychology*, 4:226.

Shervin Malmasi and Mark Dras. 2018. Native language identification with classifier stacking and ensembles. *Computational Linguistics*, 44(3):403–446.

Shervin Malmasi, Keelan Evanini, Aoife Cahill, Joel Tetreault, Robert Pugh, Christopher Hamill, Diane Napolitano, and Yao Qian. 2017. A report on the 2017 native language identification shared task. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 62–75, Copenhagen, Denmark. Association for Computational Linguistics.

Ilia Markov, Vivi Nastase, and Carlo Strapparava. 2018a. Punctuation as native language interference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3456–3466, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ilia Markov, Vivi Nastase, Carlo Strapparava, and Grigori Sidorov. 2018b. The role of emotions in native language identification. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 123–129, Brussels, Belgium. Association for Computational Linguistics.

Ella Rabinovich, Yulia Tsvetkov, and Shuly Wintner. 2018. Native language cognate effects on second language lexical choice. *CoRR*, abs/1805.09590.

Björn W. Schuller, Stefan Steidl, Anton Batliner, Julia Hirschberg, Judee K. Burgoon, Alice Baird, Aaron C. Elkins, Yue Zhang, Eduardo Coutinho, and Keelan Evanini. 2016. The INTERSPEECH 2016 computational paralinguistics challenge: Deception, sincerity & native language. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 2001–2005, San Francisco, California, USA. ISCA.

Philipp Singer, Fabian Flöck, Clemens Meinhart, Elias Zeitfogel, and Markus Strohmaier. 2014. Evolution of Reddit: From the front page of the internet to a self-referential community? In *Proceedings of the 23rd International Conference on World Wide Web*, pages 517–522, Seoul, Korea. ACM.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE 2.0: A continual pre-training framework for language understanding. In *34th AAAI Conference on Artificial Intelligence*, pages 8968–8975, New York, New York, USA. AAAI.

Joel Tetreault, Daniel Blanchard, and Aoife Cahill. 2013. A report on the first native language identification shared task. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 48–57, Atlanta, Georgia. Association for Computational Linguistics.

Joel Tetreault, Daniel Blanchard, Aoife Cahill, and Martin Chodorow. 2012. Native tongues, lost and found: Resources and empirical evaluations in native language identification. In *Proceedings of COLING 2012*, pages 2585–2602, Mumbai, India. The COLING 2012 Organizing Committee.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS 2017)*, pages 5998–6008, Long Beach, California, USA. NeurIPS.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big Bird: Transformers for longer sequences. *CoRR*, abs/2007.14062.