BigNLI: Native Language Identification with Big Bird Embeddings

Sergey Kramp, Giovanni Cassani, Chris Emmery
CSAI, Tilburg University
sergey.kramp@gmail.com, g.cassani@tilburguniversity.edu, cmry@pm.me

Abstract

Native Language Identification (NLI) intends to classify an author's native language based on their writing in another language. Historically, the task has heavily relied on time-consuming linguistic feature engineering, and NLI transformer models have thus far failed to offer effective, practical alternatives. The current work shows input size is a limiting factor, and that classifiers trained using Big Bird embeddings outperform linguistic feature engineering models (for which we reproduce previous work) by a large margin on the Reddit-L2 dataset. Additionally, we provide further insight into input length dependencies, show consistent out-of-sample (Europe subreddit) and out-of-domain (TOEFL-11) performance, and qualitatively analyze the embedding space. Given the effectiveness and computational efficiency of this method, we believe it offers a promising avenue for future NLI work.

Keywords: natural language identification, transformer embeddings, stylometry, text classification

1. Introduction

Native Language Identification (NLI) operates under the assumption that an author's first language (L1) produces discoverable patterns in a second language (L2) (Odlin, 1989; MacDonald, 2013). Classifying one's native language proves highly useful in various applications, such as in language teaching, where customized feedback could be provided based on the learner native language; in fraud detection, where identifying an unknown author's native language can aid in detecting plagiarism and web fraud; and in consumer analytics. NLI models historically relied on handcrafted linguistic patterns as input features (Koppel et al., 2005; Tetreault et al., 2013; Cimino et al., 2013; Chen et al., 2017); however, such representations are unlikely to capture all required nuances and complexities of this task (Moschitti and Basili, 2004), in particular on noisier sources of data.

Current transformer models (Vaswani et al., 2017) have shown success in such challenges (Brown et al., 2020) but are often limited by input size. This is particularly problematic for NLI which often deals with long texts, such as essays, documents or social media posts. Our work[1] is the first to employ long-form transformer models to overcome these task limitations. We train a simple logistic regression classifier which only uses the embeddings from a (fine-tuned) Big Bird (Zaheer et al., 2020) model as input, and demonstrate it significantly outperforms a similar classifier trained using costly handcrafted feature representations, at a fraction of the inference time. In our analyses, we show largely consistent out-of-sample, and out-of-domain performance, and that the embeddings encode linguistic patterns relevant to NLI.

2. Related Work

Seminal NLI work by Koppel et al. (2005) used function words, character n-grams, and handcrafted error types as features extracted from 1000 articles in five languages. The TOEFL-11 dataset (Blanchard et al., 2013) proved a fruitful resource for two NLI shared tasks (Tetreault et al., 2013; Malmasi et al., 2017). However, its controlled collection environment and limited range of topics affected generalization of traditional linguistic features to noisy Internet data (Baldwin et al., 2013). An example of such noisy data is the Reddit-L2 dataset (Rabinovich et al., 2018); the current de facto benchmark for NLI, which we employ as well.

Despite various attempts using neural architectures (Ircing et al., 2017; Bjerva et al., 2017; Franco-Salvador et al., 2017), the current best performance on the Reddit-L2 dataset was obtained by Goldin et al. (2018) using a logistic regression classifier trained on a combination of linguistic features. We implement, and thereby directly compare to, their work in our experiments.

Most related to the current work are two studies using transformers for NLI. Steinbakken and Gambäck (2020) fine-tuned BERT (Devlin et al., 2019) on a less challenging split[2] of the Reddit-L2 dataset, applying the model stand-alone, and in an ensemble of classifiers. Lotfi et al. (2020) fine-tuned GPT-2 (Radford et al., 2019) per language in the TOEFL-11 dataset, classifying a test instance according to the language-specific model with the lowest loss on that instance. Our method offers a stand-alone transformer model approach with a much lower computational footprint. We will evaluate performance on the Reddit-L2 split with little to no information related to (linguistic) geography.

[1] Code, data snapshots, model weights, and experimental details are available at https://github.com/SergeyKramp/mthesis-bigbird-embeddings.

[2] This split only includes Europe-themed subreddits, the content of which frequently reveals the author's geographical location through (e.g.) named entities.

LREC-COLING 2024, pages 2375–2382
20-25 May, 2024.

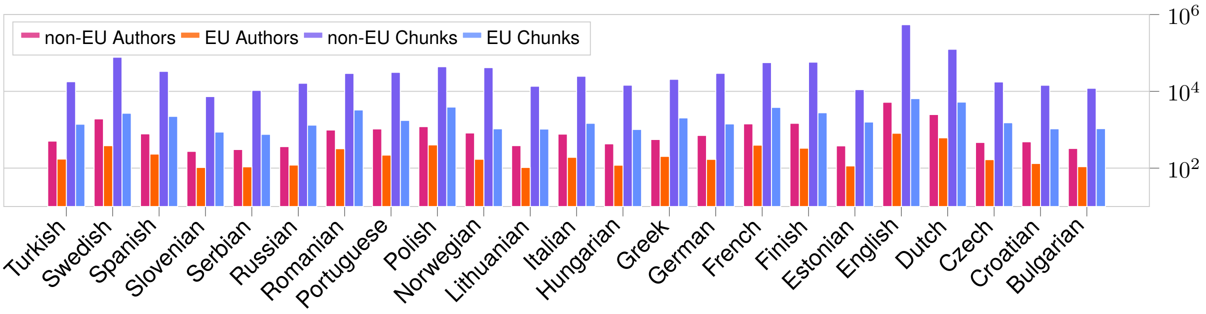

Figure 1: Logscale author and chunk frequencies per L1 in the `europe` and `non-europe` partitions.

Rabinovich et al. (2018) have used hierarchical clustering to investigate the relationship between an author's native language and their lexical choice in English. Using word frequency and embeddings of English words, they measured distances between 31 L1s, showing that languages from the same family appear closest in a vector space. They further suggested that authors with a similar L1 have comparable idiosyncrasies in their English writing. Hence, given an accurate model, we expect to observe such patterns in the embeddings used in the current work as well.

3. Methodology

3.1. Data

We used a derivative of the Reddit-L2 dataset,[3] first introduced as L2-Reddit by Rabinovich et al. (2018), and used in Goldin et al. (2018). Data collection of 200M sentences (∼3B tokens) from 2005-2017 used flairs that report country of origin on subreddits discussing European politics, yielding a total of 45K labeled native and non-native English-speaking users and their entire post history. Between-group language proficiency was accounted for through several syntactic and lexical metrics, and languages with fewer than 100 authors were removed. Each author profile was split per 100 sentences, and these "chunks" were subsequently divided in two: one partition with subreddits discussing European politics (referred to as the `europe` partition), and a second partition from all other subreddits (the `non_europe` partition). Figure 1 visualizes the partition frequencies.

Sampling For L1 identification, we regrouped Reddit-L2 on native language rather than nationality. After filtering predominantly multi-lingual countries, this resulted in 23 labels. We found that the majority are native English speakers, followed by Dutch, and that there is a stronger label imbalance in the `non_europe` partition than in `europe`.

In accordance with Goldin et al. (2018), the data was balanced through downsampling by randomly selecting 273 and 104 authors respectively (based on their least represented language) for each language in the two partitions. The amount of chunks per author was capped to reduce activity skew. These were randomly sampled up to the median per author; 17 for `non_europe`, and 3 for `europe`.

Preprocessing For this, we removed redundant blank spaces and replaced all URLs with a special token. While minimal, these changes improved classification performance across the board.

Splitting We split the `non-europe` partition on chunk level[4] into equal fine-tuning (D_{tune}), and training and testing (D_{exp}) parts. A given author might be represented in multiple chunks; hence, we did not shuffle before splitting. We hypothesized that due to the size and variety of the `non_europe` partition, it is a more realistic, challenging part of the data. Unlike the `europe` partition used by Steinbakken and Gambäck (2020), it covers a variety of topics and contains fewer context words (e.g., countries and nationalities) that might pollute classification. Instead, we dedicated the entire `europe` partition to conduct an out-of-sample evaluation. We refer to this data as D_{oos}. As this part of the data contains texts on topics not seen in D_{tune} and D_{exp}, this allows us to gauge the context specificity of our representations.

3.2. Feature Engineering Baseline

The linguistic features[5] (5186 total) were constructed following Goldin et al. (2018) (or using close equivalents), and extracted for each chunk:

n-Grams To extract the 1000 most common unigram and character tri-gram features, we used `scikit-learn` (Pedregosa et al., 2011) vectorizers fit on the text chunks of D_{exp}.

[3] Via: `http://cl.haifa.ac.il/projects/L2/`

[4] Splitting by authors had negligible effects.

[5] For comparison sake, we did not optimize these.

Edit Distance & Substitution For each misspelled word (identified using symspellpy[6]) in D_{exp}, we obtained its closest correction with a maximal edit distance of 2. Words for which no correction was found were ignored. The required insertions, deletions, and replacements formed a substitution frequency list, of which the top 400 were used as features. Additionally, for each chunk we summed the Levenshtein distance between all words and their corrections, divided by the total number of words, giving the average edit distance.

Miscellaneous To extract all other features, each chunk in D_{exp} was first split by \n. Binary grammar error features (i.e., the presence or absence an an error in that chunk) were extracted using LanguageTool[7] (2017 error types in total). The top 300 POS tri-grams were extracted with nltk[8] (Wagner, 2010), and function word frequency features used a list (Volansky et al., 2015, 467 total). For average sentence length, we removed all non-alphanumeric symbols of length 1, then divided sentence length (on word level) by the total number of sentences in a chunk (i.e., 100).

3.3. Transformer Model

To efficiently apply transformers for NLI we opted for Big Bird (google/bigbird-roberta-base on the Hugging Face Model Hub; Paszke et al., 2019; Wolf et al., 2019) as it has a relatively large context length of 4096 tokens while fitting on a single GPU.[9]

Fine-tuning We fine-tuned all layers of Big Bird on D_{tune} using the hyperparameters specified in the original paper: Adam (Kingma and Ba, 2015) to optimize with the learning rate set to 10^{-5} and epsilon to 10^{-8}. Warm-up on 10% of all training inputs ran during the first epoch. Fine-tuning ran for 3 epochs totaling 15 hours. Due to memory constraints, we used an input size of 2048, with a batch size of 2. Chunks that were shorter were padded to match the input length; longer inputs were split into sub-chunks (padded to full length).

Embedding Representation In order to compare Big Bird to linguistic features, we only extract its embeddings (either pre-trained from the Model Hub or our own fine-tuned version), using them as input to a classifier. Per chunk, we added [CLS] at the beginning of the first sentence, and manually inserted a separator token between each sentence and at the end of the chunk. We then used the last hidden states for [CLS] as the chunk's 768-dimensional embedding features. We experimented with 3 token input sizes: 512 (BERT's input size), 2048 (size also used when fine-tuning), and 4096 (Big Bird's maximum input size).

4. Experimental Setup

4.1. Main Experiment

We trained a logistic regression classifier on the output of each feature extractor. To further establish an equal ground for comparison, we did not tune the hyperparameters of these classifiers. Hence, we adopted scikit-learn's default parameters: ℓ_2 normalization, $C = 1$, L-BFGS (Liu and Nocedal, 1989) for optimization, and maximum iterations set to 1000. We used average accuracy over 10-fold cross-validation (CV) to gauge the robustness of each classifier's performance.

4.2. Embedding Space Analysis

Following Rabinovich et al. (2018), we used hierarchical clustering to analyze how each native language is represented in the 768-dimensional embedding space. We used the best performing pre-trained and fine-tuned Big Bird models from our main experiment to compute the centroids (23 in total) on D_{exp}. Subsequently, we used scipy's (Virtanen et al., 2020) implementation of Ward's linkage function (Ward, 1963) to create a cluster dendrogram, and scikit-learn's default implementation of Principal Component Analysis (Girshick, 1936; Tipping and Bishop, 2002, PCA) to visualize the centroids in a 2-D space.

4.3. Error Analysis

Out-of-Sample (OOS) Analysis To assess generalization,[10] we trained three classifiers (one per representation method) on D_{exp} and tested on D_{oos} (only concerns European politics; generally absent in D_{exp}). Our baseline uses linguistic features, and two classifiers use Big Bird embeddings from the best performing pre-trained and fine-tuned feature extractors (see Table 1). We considered both versions of the feature extractor to control for any data leakage that occurred during fine-tuning.

Sensitivity to Text Length To gauge the effect of text length on performance, we randomly sampled 1000 chunks from D_{exp} and created slices[11] of 10%, 20%, 40%, and 80% of the total length of the chunk, following a similar baseline and embedding

[6] github.com/mammothb/symspellpy

[7] github.com/jxmorris12/language_tool_py

[8] We used the pre-trained Averaged Perceptron Tagger in combination with the Punkt Tokenizer.

[9] We used an NVIDIA Titan X with 12 GB of VRAM.

[10] Big Bird was reportedly not trained on Reddit.

[11] Sliced on \n. We also experimented with sentence, clause, and character-level—all yielding similar results.

Name	Hours	ACV	OOS	OOD
Feature Eng.	13.00	.475	.637	.172
BigBird-512	0.27	.364	-	-
BigBird-512-t	0.27	.432	-	-
BigBird-2048	2.50	.493	.774	.102
BigBird-2048-t	2.50	**.654**	**.855**	**.204**
BigBird-4096	3.00	.500	-	-
BigBird-4096-t	3.00	.635	-	-

Table 1: The models (Name) annotated with their input dimensions and if they were fine-[t]uned, how long feature extraction took on D_{exp} (Hours), their average cross-validation accuracy scores on D_{exp} (ACV) and accuracy scores on D_{oos} (OOS, `r/Europe`) and D_{ood} (OOD, TOEFL-11).

extraction method as the out-of-sample analysis. Next, we trained a logistic regression classifier, similar to those described in Section 4.1, on all of D_{exp} except the 1000 randomly sampled chunks. Then, we obtained predictions for all slices, and computed the accuracy for each slice group.

Out-of-Domain (OOD) Analysis In order to measure true out-of-domain performance, we used the TOEFL-11 (Blanchard et al., 2013) set as D_{ood}; specifically the test split, filtered on the five languages that overlap with our training data (French, German, Italian, Spanish, and Turkish). It should be noted that the average amount of tokens per instance for TOEFL (322) is significantly lower than the average in D_{exp} (1726). Hence, we expect performance to suffer as a result.

5. Results

5.1. Main Experiment

Table 1 shows the average CV scores of each classifier. BigBird-2048-t yielded the highest average CV accuracy with 65.38%; a 17 point increase over the baseline trained on linguistic features (47.55%). The classifiers trained on fine-tuned embeddings outperformed those using pre-trained embeddings across all three model variants. However, differences are smallest for BigBird-512, suggesting that the short input size limits fine-tuning's efficiency. Increasing input size beyond 2048 tokens seems to have a small effect; however, given that the average chunk length in D_{exp} is 1726 tokens, with an input size of 2048 tokens, most are captured already.

Finally, our classifiers show comparable errors between L1s; in most cases, the classifiers confuse the true language with a language from the same language family or a language of a nearby country (e.g. Serbian with Croatian, Croatian with Russian or Polish with Czech).

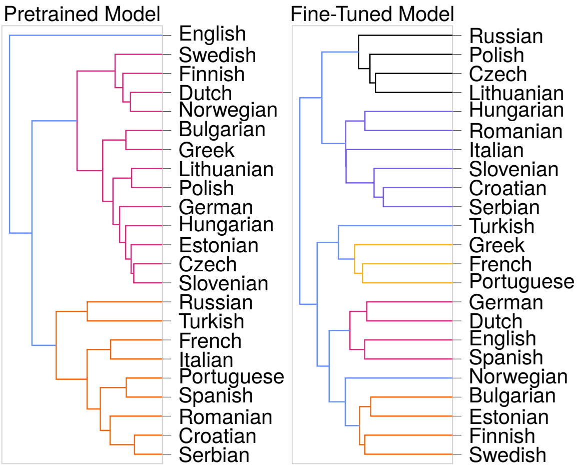

Figure 2: Hierarchical clustering dendograms of native language centroids in the Big Bird embedding space before and after fine-tuning.

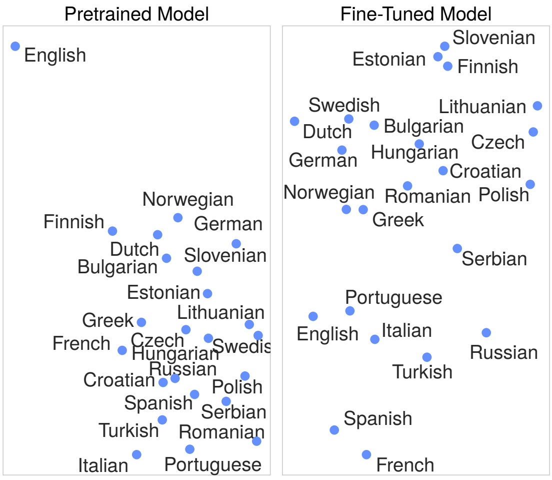

Figure 3: 2-Dimensional PCA space showing the language centroids before and after fine-tuning.

5.2. Embedding Space Analysis

Although our clustering shows some overlap with the results of Rabinovich et al. (2018), there are some deviations. Languages from the same language family are not always close (see Figure 2, fine-tuned or not). For example, Russian is clustered with Turkish (pre-trained) and Italian with the former Yugoslavian languages (fine-tuned). Furthermore, fine-tuning shifts the embedding space more toward separating individual languages, rather than separating native-English from non-native English (as indicated by English having its own cluster). This effect is most apparent in the low-dimensional PCA space (see Figure 3). In the fine-tuned space, an interesting artifact can be observed, where the space roughly mimics the languages' geographical orientation to each other.

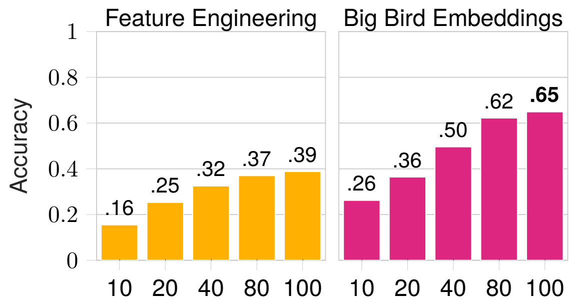

Figure 4: Baseline and embedding model accuracy scores by percentage of total input length.

5.3. Error Analysis

Out-of-sample Analysis Here we see the same pattern as in our main experiment (see Table 1), with the fine-tuned embedding approach yielding the most accurate classifier, outperforming the feature engineering baseline by 22 percentage points, whereas the pre-trained model gains 13.7.

Sensitivity to Text Length In Figure 4, it can be observed that the performance of both embedding and feature engineering classifiers deteriorates as text length decreases. However, the deterioration is not linear, which suggests there is increased redundancy in the information used for classification the longer the input becomes. The embeddings are more affected, with a 12 point drop when reducing from 80% to 40% and a 14 point drop when reducing from 40% to 20%, compared to 5 points and 7 points for the feature engineering model.

Out-of-Domain Analysis Turning to the results in Table 1 again, we can observe a strong drop-off in performance when both feature engineering and embedding-based models are applied to shorter, closed-form text. Interestingly, with the average TOEFL document being 15.3% of the maximum input length, the performance is only slightly lower than the expected in-domain performance under such input constraints (see Figure 4 for comparable input length effects). Note that this only provides a contextual view on performance differences; TOEFL-11 models' benchmark performance is close to 90% accuracy (Malmasi et al., 2017).

Cross-Domain Analysis Results of the previous error analyses called for further cross-examination (reported in Table 2).[12] Here, in addition to the decreased Reddit-L2 subset difficulty with fewer labels, we can observe the same performance patterns—with one exception. Without additional optimization, the feature engineering model seems more suited for TOEFL (while not included here, our models evenly score ∼10% less on the full TOEFL-11 task). BigBird-2048-t also seems to achieve slightly better out-of-domain performance when trained on TOEFL; with a performance drop of 25.7% on non-europe, compared to 33.9% the other way around. However, better performance on TOEFL also seems to cause poorer out-of-domain generalization. This might suggest this benchmark may cause overfitting. Further investigation in this cross-domain setting, in particular featuring previous implementations tested on TOEFL-11 (Malmasi et al., 2017), would certainly be a worthwhile contribution to future NLI work.

Train	~~EU~~		TFL	
Test	~~EU~~	TFL	~~EU~~	TFL
Feature Eng.	.729	.262	.406	**.754**
BigBird-2048	.748	.280	.312	.660
BigBird-2048-t	**.821**	**.370**	**.610**	.560

Table 2: Cross-evaluation accuracy scores between different models trained and tested on the non_europe (~~EU~~) and TOEFL-11 (TFL) datasets.

6. Discussion & Conclusion

Our experiments demonstrate how fairly straightforward feature extraction using embeddings from transformers that account for long enough input sequences is faster, and substantially outperforms prior best performing models on Reddit-L2. Some limitations should be mentioned here: the domain is rather restricted, as Reddit's demographics imply the dataset mostly contains highly fluent English speakers, which, in turn, was also the only L2 we focused on. Hence, other social platforms are worth evaluating on as well (although label collection will likely be significantly more challenging).

For future work, we expect even better results might be achieved tuning other classifiers than logistic regression, and a comparison with similar transformers such as Longformer (Beltagy et al., 2020) and Transformer-XL (Dai et al., 2019) is certainly worthwhile (Bulatov et al., 2023). As is commonly observed (Devlin et al., 2019; Sun et al., 2019; Howard and Ruder, 2018), fine-tuning Big Bird on our data improved performance, and our observations proved robust both throughout cross-validation and on out-of-sample data. Given the results and error analyses, we believe our works offers various starting points for future NLI work, and that the ideas presented may be broadly applied as an efficient method in text classification problems that specifically deal with longer inputs.

[12] Please note that these experiments used a train/test split with identical labels (five languages per set); hence, these experiments are markedly different from Table 1.

7. Acknowledgments

Our research strongly relied on openly available resources. We thank all whose work we could use. We would also like to thank a subset of the reviewers for their helpful comments.

8. Bibliographical References

Timothy Baldwin, Paul Cook, Marco Lui, Andrew MacKinlay, and Li Wang. 2013. How noisy social media text, how diffrnt social media sources? In *Sixth International Joint Conference on Natural Language Processing, IJCNLP 2013, Nagoya, Japan, October 14-18, 2013*, pages 356–364. Asian Federation of Natural Language Processing / ACL.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Johannes Bjerva, Gintarė Grigonytė, Robert Östling, and Barbara Plank. 2017. Neural networks and spelling features for native language identification. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 235–239, Copenhagen, Denmark. Association for Computational Linguistics.

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.

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. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Aydar Bulatov, Yuri Kuratov, and Mikhail S. Burtsev. 2023. Scaling transformer to 1m tokens and beyond with RMT. *CoRR*, abs/2304.11062.

Lingzhen Chen, Carlo Strapparava, and Vivi Nastase. 2017. Improving native language identification by using spelling errors. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, pages 542–546. Association for Computational Linguistics.

Andrea Cimino, Felice Dell'Orletta, Giulia Venturi, and Simonetta Montemagni. 2013. Linguistic profiling based on general-purpose features and native language identification. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications, BEA@NAACL-HLT 2013, June 13, 2013, Atlanta, Georgia, USA*, pages 207–215. The Association for Computer Linguistics.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2978–2988. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Marc Franco-Salvador, Greg Kondrak, and Paolo Rosso. 2017. Bridging the native language and language variety identification tasks. *Procedia Computer Science*, 112:1554–1561.

M. A. Girshick. 1936. Principal components. *Journal of the American Statistical Association*, 31(195):519–528.

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.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20,* *2018, Volume 1: Long Papers*, pages 328–339. 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.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Moshe Koppel, Jonathan Schler, and Kfir Zigdon. 2005. Automatically determining an anonymous author's native language. In *Intelligence and Security Informatics, IEEE International Conference on Intelligence and Security Informatics, ISI 2005, Atlanta, GA, USA, May 19-20, 2005, Proceedings*, volume 3495 of *Lecture Notes in Computer Science*, pages 209–217. Springer.

Dong C. Liu and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Math. Program.*, 45(1-3):503–528.

Ehsan Lotfi, Ilia Markov, and Walter Daelemans. 2020. A deep generative approach to native language identification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1778–1783, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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

Shervin Malmasi, Keelan Evanini, Aoife Cahill, Joel R. Tetreault, Robert A. 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, BEA@EMNLP 2017, Copenhagen, Denmark, September 8, 2017*, pages 62–75. Association for Computational Linguistics.

Alessandro Moschitti and Roberto Basili. 2004. Complex linguistic features for text classification: A comprehensive study. In *Advances in Information Retrieval, 26th European Conference on IR Research, ECIR 2004, Sunderland, UK, April 5-7, 2004, Proceedings*, volume 2997 of *Lecture Notes in Computer Science*, pages 181–196. Springer.

Terence Odlin. 1989. *Language Transfer*. Cambridge Applied Linguistics. Cambridge University Press.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830.

Ella Rabinovich, Yulia Tsvetkov, and Shuly Wintner. 2018. Native language cognate effects on second language lexical choice. *Trans. Assoc. Comput. Linguistics*, 6:329–342.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Stian Steinbakken and Björn Gambäck. 2020. Native-language identification with attention. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 261–271, Indian Institute of Technology Patna, Patna, India. NLP Association of India (NLPAI).

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune BERT for text classification? In *Chinese Computational Linguistics - 18th China National Conference, CCL 2019, Kunming, China, October 18-20, 2019, Proceedings*, volume 11856 of *Lecture Notes in Computer Science*, pages 194–206. Springer.

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.

Michael E. Tipping and Christopher M. Bishop. 2002. Probabilistic Principal Component Analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 61(3):611–622.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Vered Volansky, Noam Ordan, and Shuly Wintner. 2015. On the features of translationese. *Digit. Scholarsh. Humanit.*, 30(1):98–118.

Wiebke Wagner. 2010. Steven bird, ewan klein and edward loper: Natural language processing with python, analyzing text with the natural language toolkit - o'reilly media, beijing, 2009, ISBN 978-0-596-51649-9. *Lang. Resour. Evaluation*, 44(4):421–424.

Joe H. Ward, Jr. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.