Faculty of Arts

Master's Thesis

Digital Text Analysis

Native Language Identification of Greek in English-Written Texts

A Comparison of Machine Learning Approaches

Dimitrios Boumparis

Supervisor: Prof. Dr. Walter Daelemans

Co-supervisor: Jens Lemmens

Assessor: Jens Van Nooten

Academic year 2022–2023

The undersigned, Dimitrios Boumparis, student of the Master program in Digital Text Analysis at the University of Antwerp, declares that this thesis is completely original and exclusively written by himself. For all information and ideas derived from other sources, the undersigned has referred to the original sources, both explicitly and in detail.

Antwerp, 08/01/2023
Dimitrios Boumparis

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Prof. Dr Walter Daelemans, for his guidance and support throughout this research project. His expertise and encouragement have been truly invaluable. I would also like to thank PhD student Jens Lemmens for his insightful advice and for being a great colleague. I am also grateful to all the professors I have had the opportunity and honour of working with during my academic journey so far. Their believing in me and mentorship have contributed significantly to my personal and professional development. I would like to give special mention to Dr George Mikros, Dr John Karras, and Dr Andreas Giannakoulopoulos for their unwavering support. Their support has improved the quality of my work. Their unconditional belief in my potential has inspired me to strive for excellence in all my endeavours. Finally, I am deeply grateful to my family and friends for their constant love and support. They have always been the driving force behind everything I have accomplished to date. Thank you all for being such an integral part of my growth.

Native Language Identification of Greek in English-Written Texts

A Comparison of Machine Learning Approaches

Dimitrios Boumparis
University of Antwerp
Faculty of Arts
Master in Digital Text Analysis
dimitrios.boumparis@student.uantwerpen.be

The task of Native Language Identification aims at predicting the native language (L1) of an author given only texts produced in a foreign language (L2). English is spoken by a minority of native speakers compared to those who speak it as an L2. This means that a large volume of English content is created by non-native English speakers. Therefore, it is crucial to devise strategies for determining the native language of non-native English writers. The study at hand[1] *introduces the NLI task for native speakers of Greek with English as L2. The corpus that we used is a collection of similar-topic posts from Reddit written by native speakers of Greek in English, along with English posts written by native English speakers from the UK and the US. This study provides new insights into the challenges and opportunities of identifying languages in multilingual contexts but also compares the performance across a variety of features of linguistic nature and employing both statistical and neural algorithms to shed light on how the two approaches differ in performance while tackling the NLI task. It lays the first stones to introduce the NLI task to other L1s in the future, train classifiers and check their robustness by finding that the combination of word embeddings with simple n-gram features on characters and Universal Parts-of-Speech tags achieves 82.6% mean F1-macro score after 10-fold Cross-Validation, surpassing most previous NLI studies. In addition, our best pre-trained BERT-based approach achieves equally high results (82.1%). A hard-voting ensemble model was also trained by combining three classifiers with different linguistic and word-embeddings features, however without resulting in performance increase. The error analysis conducted thereafter suggests that the short-form nature of the posts and the common usage of words that are mainly found in the opposite class could be some of the main reasons that these results could be attributed to. Finally, an important finding could potentially indicate that L1 word etymology could relate to L2 lexical choices, which would make an attractive case for upcoming research in the field of NLI.*

Keywords: Computational Linguistics, Native Language Identification, Support Vector Machines, Word embeddings, Pre-trained Large Language Models.

[1] Data and code available on https://github.com/dimboump/GreekNLI.

Contents

List of Abbreviations

AI Artificial Intelligence

BERT Bi-directional Encoding Representation from Transformers

CLI cross-linguistic influence
CNN Convolutional Neural Network
CV Cross-Validation

DL Deep Learning

FFNN Feed-Forward Neural Network

ICLE International Corpus of Learner English

L1 native language
L2 non-native language
LDA Latent Dirichlet Allocation
LLM large language model
LSTM Long Short-Term Memory
LUD language of unlimited diffusion

ML Machine Learning

NER Named Entity Recognition
NLI Native Language Identification
NLP Natural Language Processing
NMT Neural Machine Translation
NN Neural Network
NNS non-native speaker
NS native speaker

RNN Recurrent Neural Network

SGD Stochastic Gradient Descent
SLA Second Language Acquisition
SVM Support Vector Machines

TF-IDF Term Frequency-Inverse Document Frequency
TOEFL Test of English as a Foreign Language
TSG Tree Substitution Grammar

1 Introduction

1.1 What is Native Language Identification?

Native Language Identification (NLI) is the task in Natural Language Processing (NLP) of predicting the native language (L1) of an author based on texts written in a language other than the author's native (L2). It is based on the hypothesis that speakers sharing the same L1 use common language patterns when writing in L2s and tend to choose more similar words to express the same or similar concepts. In literature, this is known as cross-linguistic influence (CLI) (Malmasi and Dras 2017). For example, native speakers of Russian will tend to use *the* less in their L2 English than a native English speaker, because the notion of articles is not present in Russian (Tsur and Rappoport 2007). NLI can be also applied in oral speech (Yarra, Rao M.V., and Kumar Ghosh 2018), albeit this falls outside the scope of this study. Figure 1 below gives an overview of the NLI task.

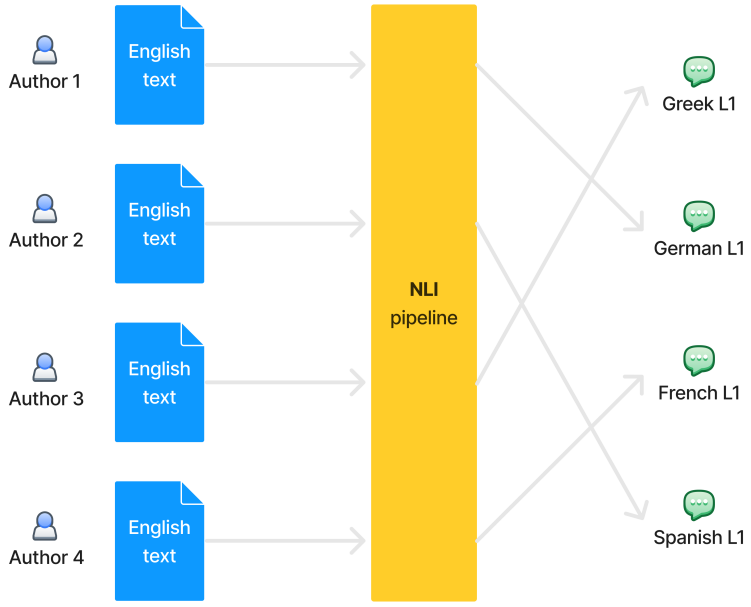


Figure 1: An overview of the NLI task, i.e., trying to classify the native languages (L1s) of authors based solely on texts written in non-native English (L2) (adapted from Malmasi and Dras 2017, 164).

Native Language Identification is a field of study with many practical applications. One key area where it is useful is in Second Language Acquisition (SLA). By analysing linguistic patterns in the writing of learners of foreign languages, NLI can help to provide constructive feedback that targets specific areas where they may be struggling. For example, this can be especially useful for learners who are trying to overcome errors that are characteristic of their native language but not appropriate in the second language.

Another area where NLI has found significant use is in Forensic Linguistics. In a law enforcement setting, it is often important to identify the linguistic background of an anonymous author of a text. NLI can be used, in combination with other NLP tasks and techniques, to help profile the author with regard to the linguistic background, which can be an essential piece of information in a forensic investigation.

1.2 Greek and the NLI landscape

NLI is arguably not the most common task in the NLP field. The primary reason is that it requires a lot of stars to align before it can be tackled. In the case of NLI, models cannot simply focus on specific words but, rather, try to extract deeper patterns that may lay behind them, such as syntax. As we will discuss in the Section 2.1, there are many other factors that need to be addressed prior to compiling a dataset for NLI. Usually, issues emerge with regard to gender and age balance, but the literature list topic neutrality as a priority. Apart from the text being independent in terms of topic, a more practical issue lies in the availability of NLI-ready data and, by extension, verifying that it is suitable for use. To be more concrete, texts written in L2 English by non-native speakers are hard to find, especially when the primary source of data is the internet where anonymity prevails, and it is a true challenge to verify that the author is not an NS of the language in question. To add more complexity to this challenge, the collector of the data should go one step further and assign the class of the author's L1 as well. This makes the collection of texts a hard mission, discourages researchers from tackling the NLI task, and makes the latter difficult to undertake at a large scale compared to other NLP tasks.

This has resulted in only *major languages* (or languages of unlimited diffusion (LUDs), as Radó 1987, 6-7 calls them)) having been explored in NLI tasks. A "major" language is defined as a 'world language', namely languages that are spoken by tens or hundredths of millions of people as their mother tongue but also are not limited within the territory of their country. Not only that, but there is an even more limited amount of L2s having been explored other than English as part of NLI. In such cases, we are dealing with what Malmasi and Dras (2017) call *Multilingual NLI*. They were the first to carry out a handful of experiments using six L2s other than English: Italian, German, Spanish, Chinese, Arabic, and Finnish. The present study does not delve into multilingual NLI, it is nevertheless interesting to see what the NLI landscape looks like in terms of languages.

At the time of writing, no NLI research has been done between Greek and English as L1 and L2, respectively. This makes the present study a first in this regard.

1.3 Comparison of Machine Learning Approaches

The task of predicting the native language of an author was first introduced into the field of Natural Language Processing (NLP) by Koppel, Schler, and Zigdon (2005a), who also set the bar very high from the very start (see Section 2). They established the classification paradigm in NLI which the literature has been following since.

However, the advent of Neural Networks (NNs) in early and mid-2010s, changed the landscape in Artificial Intelligence (AI) and redefined what is possible in language modelling, among other fields. Slowly but surely, new accuracy highs were reached, and the state-of-the-art has started moving faster than ever. Starting from Simple Feed-Forward Neural Networks (FFNNs), progressing further to Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) along with their evolution, the

Long Short-Term Memory (LSTM), and, most recently, the Attentive NNs (ANNs), the field of language modelling has seen unprecedented performance gains, e.g., in Neural Machine Translation (NMT), and reached or even outperformed human-level output in many NLP tasks, such as NLI, Named Entity Recognition (NER), and question and answering (Goldberg and Hirst 2017). The most credit for this achievement deserves to be given to two recent advancements in the NLP space: (a) the Transformer NN architecture, introduced by Vaswani et al. (2017), and (b) and pre-trained, large language models (LLMs), introduced by Devlin et al. (2018) with the Bi-directional Encoding Representation from Transformers (BERT). We will be comparing the performance of our two approaches, as part of the Greek NLI task, the first of which involves two statistical pipelines using a Support Vector Machines (SVM) and a Logistic Regression classifier, and the second will utilise two fine-tuned pre-trained models based on BERT.

It is very interesting to compare how these two approaches deal with the NLI task, not in terms of composition but rather in terms of feature extraction for the former and simply input data for the latter. As we will see in the next section (Section 2), the NLI task has traditionally been a classical Machine Learning (ML) problem. So much indeed, that no Deep Learning (DL) approach has secured the winning place on the 2017 Shared Task (since no team participated with a DL approach in the 2013 Shared Task) and the literature regarding such approaches is fairly limited, relative to ML. Some reasons as to why this happened could be (a) that there are other, more "appealing" tasks that were first introduced to the neural networks universe for people to see how this novel methods deal with; (b) the previous NLI research had already achieved very high scores; and (c) the available data to model are very rare and difficult to compile (see Section 3.1) Finally, in Section 3, we will analyse the models that we will be comparing, after we first refer to the data used in both settings.

1.4 Contribution

As previously mentioned, the study at hand has a two-fold aim. First, it introduces Greek in the NLI task of predicting the L1 of an author with English as L2, which – to the best of our knowledge – has not been done before. Second, it attempts to examine different feature sets and classifiers involving statistical and neural models, combined with linguistic features and word embeddings.

1.4.1 Research questions

To this end, and with all the above in mind, this study will attempt to give a satisfactory answer to the following questions:

RQ1: Is it possible to distinguish English texts written by Greek native speakers from native speakers of other languages using learning algorithms?
RQ2: How does the performance of statistical machine learning approaches compare to more recent deep learning approaches?

How we go about answering the aforementioned questions will be discussed in detail in Section 3.

1.4.2 Limitations of the study

However, before diving into the methodology used to tackle the problem Section 3 and discuss the results (Section 4), we first need to enumerate the limitations that come with a quantitative study. Without a doubt, the main limitation is the sample size. As we will see in Section 2.1, when discussing about the NLI task, sample size has always been a difficult challenge to overcome, especially in comparative studies dealing with an increasing number of L1 languages (e.g., Malmasi and Dras 2017). Another limitation that is inherent to such studies is the generalisation problem. In the case of this paper, a corpus including short to mid-sized posts from a popular forum-like website was used for training the classifiers. It is generally true that posts in social media do not conform to the norms of the written speech, especially with regard to punctuation and letter casing, and therefore it is very challenging to generalise, especially when the topics of the texts are not identical or similar enough. More details about the data and the methodology are given in Section 3. Finally, one other limitation of this study is the potential for bias to affect the validity and reliability of the results. Bias can occur at various stages of the research process, including the selection of the sample, the measurement of variables, and the analysis of data. For example, if the sample is not representative of the population being studied, the results may not be able to generalise to the larger population. As we will see in Section 3.1, the data come from an online source where metadata such as age and gender are not disclosed. The same applies for L2 language proficiency. We therefore cannot verify the distribution of our samples with regard to these variables. It is, finally, worth mentioning that only a few topics could be present in the texts while others are not represented enough or even at all. This would essentially prevent the models from performing similarly to the results reported in Section 4 in out-of-domain unseen data.

2 Related work

The 2013 NLI Shared Task played a significant role in popularising the NLI task (Tetreault, Blanchard, and Cahill 2013) along with its second iteration in 2017 (Malmasi et al. 2017), even though there were some earlier studies on the topic. This, arguably, is going to be the primary source of previous related work. Malmasi, Tetreault, and Dras (2015) have shown that automatic methods significantly outperform humans in said task (73% compared to 37% accuracy). Typically, the state-of-the-art for NLI falls within the range of 80-90% in terms of accuracy, with some studies reporting even higher results (e.g. Brooke and Hirst 2012). However, this is heavily dependent on the L1s used and the total amount of samples and topics per L1.

2.1 Corpora

The 2013 NLI Shared Task was, among other things, a crash test for the recently released TOEFL11 corpus (Blanchard et al. 2013), although it was used slightly earlier by Tetreault et al. (2012). Up until that point, there was only a single corpus available that contained English texts written by NNSs, the International Corpus of Learner English (ICLE) (Granger et al. 2009) which was primarily compiled for Corpus Linguistic purposes than NLI. The TOEFL11 corpus has become the *de facto* standard benchmark dataset for NLI.

2.1.1 ICLE

The International Corpus of Learner English (ICLE) was introduced by Granger (2003) and was later updated to version 2 (Granger et al. 2009). Since 2020, the version 3 is available (Granger et al. 2020), however most studies have been carried out using the second version of the corpus, so we will primarily focus thereon.

The ICLEv2 corpus contains around 6,000 essays with a total of 3.7 million words from 16 different language backgrounds, a significant increase over its predecessor which comprised 2.5 million words from 11 languages. The texts were written as answers to a total number of around 1,300 prompts and have been manually clustered into 736 topics. As far as the characteristics (*variables*) featured in the authors of the included texts, Granger et al. (2009, cited in Tetreault et al. 2012, 5) mention that the objective was for the texts to share a large number of commonalities. They make explicit mentions to the form (written), genre (academic essays), English register (general rather than specialised language), and length (500-1,000 words).

However, ICLE was criticised for being problematic for NLI, primarily due to topic bias (Brooke and Hirst 2013, 38-39). Tetreault et al. (2012, 5-6) suggested that little attention was given to the total amount of essays per L1. In the former case, the standard deviation σ is around 243 essays, while, in the latter case, the σ is equal to around 194 essays, with L1 Chinese taking the top spot in both cases by a whole mile over L1 Japanese which comes at the second most common language in the corpus.

2.1.2 TOEFL11

Tetreault et al. (2012) introduced a corpus from texts written for 8 prompts for the Test of English as a Foreign Language (TOEFL). The TOEFL11 corpus, as they called it, contains 11 different language background and address the bias as well as the small data size issues of the ICLE. To be specific, TOEFL11 contains 1,100 essays per L1 from 8 topics. In addition, it also provides the proficiency score of 'low', 'medium', or 'high' for each essay. Along with TOEFL11, the authors compiled a larger version, called TOEFL11-Big, to test how data size affects NLI. TOEFL11-Big consists of approximately 87,500 essays and the same 11 L1s (7,900-8,000 essays per L1). What is remarkable is that there is no overlap between the two flavours of the corpus (Tetreault et al. 2012, 7). However, this corpus has not been shared publicly. Their intention was also to directly compare the ICLEv2 and the TOEFL11 corpora.

As previously detailed in Section 2.1.1, since there is not total overlap between the languages of the two main corpora, Tetreault et al. (2012) extracted a subset thereof and named them ICLE-NLI and TOEFL7, accordingly. It is evident that the two comparable corpora now had the same number of languages (7). Prior to its compilation, ICLE-NLI was also transformed to exclude the documents headers and include only ASCII characters. The seven languages included in the corpora are Bulgarian, Chinese, Czech, French, Japanese, Russian and Spanish. Last but not least, it is important to mention that ICLE-NLI has way less essays in total (770 vs 4,613), however its essays are significantly longer on average than TOEFL7's (666 vs 252 words).

2.1.3 Other corpora

FCE corpus Introduced by Yannakoudakis, Briscoe, and Medlock (2011) to automatically assess English for Speakers of Other Languages (ESOL) essays in the context of the First Certificate in English (FCE) Test of the University of Cambridge. Apart from the

author's native language, the age and the final grade are also present. Lengthwise, the essays are 428 words long, on average. Each of the two tasks grades are marked between 1 and 40 points, and an overall mark is then assigned to each candidate student.

Lang-8 Introduced by Brooke and Hirst (2013) as a 'cheap' alternative to ICLE, since it allowed online users to submit their personal narratives through a web portal and hence considerable reducing the compilation time, conversely to ICLE which took many years to compile. The texts in Lang-8 account for 22 million words in total, with the 8 best-represented L1s that have at least 200,000 words each being: Japanese, Chinese (both Mandarin and Cantonese), Korean, Russian, Spanish, French, German, Polish, Italian, and Vietnamese.

L2-Reddit Introduced by Rabinovich, Tsvetkov, and Wintner (2018) as a large dataset of posts and comments from over 45,000 users with 46 native languages (mainly from countries with Indo-European L1s). The main source of texts were Europe-related subreddits[2], as their users are required to specify their country as a metadata attribute called *flair*. The authors (2018, 332) conducted supervised classification experiments and concluded that the specified country can be viewed as a plausible, albeit not perfect, proxy for the users' native language. They later expanded the corpus by also including posts written by the same users of the main part of the corpus in other, unrelated subreddits, accounting for more than 80,000 unique topics. For this study, we will use a subset of the L2-Reddit corpus, which we will discuss further in Section 3.1.

2.2 Approaches

The field of NLI has seen significant growth in recent years, with a variety of Machine Learning approaches being developed to tackle this problem. In this section, we will discuss the different Machine Learning and Deep Learning approaches that have been proposed for NLI and how they have been applied to said task. It is worth mentioning that all studies examined in the context of writing this section treat NLI as a classification problem, and the majority of the teams who participated in the two NLI Shared Tasks used an SVM model, which applies also for the rest of the studies mentioned in the following section.

2.2.1 Statistical models

The first study to use the ICLE corpus for NLI, namely to predict the L1 of NNSs using English texts, is Koppel, Schler, and Zigdon (2005a,b). They used the original version and, in particular, a subset thereof containing 5 L1s (Bulgarian, Czech, French, Russian, and Spanish). The features they extracted include POS tags n-grams, function and content words as well as typos and grammatical errors. The last feature was based on their suggestion that grammatical patterns present in the author's native language emerge also in L2s. Their SVM classifier achieved a very high accuracy (80.2%) considering it was the first entry on the "leader board" of the newly-born NLP task. Wong and Dras (2009) combined the features used by Koppel, Schler, and Zigdon (2005a) with

[2] *Subreddits* are the areas of interest (namely forums) into which Reddit is divided and are denoted with the prefix `r/`. The authors gathered content primarily from the Europe-related subreddits `r/Europe`, `r/AskEurope`, `r/EuropeanCulture`, `r/EuropeanFederalists`, and `r/Eurosceptics`.

syntactical errors, but the accuracy did not improve further. However, they extended the lexical model found in Koppel, Schler, and Zigdon (2005b) once again, but this time by employing more complex syntactic features, namely parse trees and context-free grammar features, and using a maximum entropy classifier. They achieved a new state-of-the-art at the time of almost 82% accuracy on the ICLE corpus (Wong and Dras 2011). Other studies explored other grammatical features, such as Tree Substitution Grammars (Swanson and Charniak 2012) and adopt grammars as an extension of Latent Dirichlet Allocation (LDA) (Wong, Dras, and Johnson 2012). Both of these studies used the ICLEv2 corpus. The authors of the former trained a Logistic Regression classifier, and the latter a Bayesian-based model which achieved accuracies of 78.4% and 75.7%, respectively.

Tetreault et al. (2012) used an ensemble of SVM classifiers and achieved very high accuracies: 90.1% on the ICLE corpus, 80.9% on the TOEFL11 and 84.6% on the TOEFL11-Big, by combining all the previous features with n-gram language models. Popescu and Ionescu (2013) ranked third on the 2013 NLI Shared Task with an accuracy of 82.7%, using a Kernel Ridge Regression (KRR), by implementing a distance measured applied on DNA sequences on the texts used for training and using n-grams of $n = 6$ and 8. When Ionescu, Popescu, and Cahill (2014) experimented with three corpora (ICLE, TOEFL11, and TOEFL11-Big) and various classifiers, combining their model with character-level features, their one-versus-all KRR and the Kernel Discriminant Analysis (KDA) classifiers yielded the highest results, placing them among the few studies examined for the purposes of this paper that did not train an SVM of any solver. A similar approach, with the addition of function words was followed by Tsur and Rappoport (2007), but their SVM model yielded 66% accuracy. Function words are a closed-set of words that carry little to no meaning – conversely to content words – however play a crucial role in forming grammatically and syntactically correct sentences. Although closed-set, the number of function words usually depends on arbitrary selection criteria[3]; Some studies use around 350 function words (Argamon, Šarić, and Stein 2003; Zhao and Zobel 2007) while others used 480 (Koppel and Schler 2003) or even 675 (Argamon et al. 2007)! What is unique about these words is they are present in every text regardless of the topic in question and, what's more, are statistically among the most common words found within them (based on Zipf's law, found in Zipf 1949, 35-37). These are the reasons why such a model is not prone to topic bias while training and function words they have been used as the only features for other NLP tasks, such as authorship attribution (Stamatatos 2009; Kestemont 2014).

Lynum (2013) trained an SVM classifier on word n-grams with $n = (1, 2)$, characters of $n = (3, 6)$, and 4 character suffix 2-grams. He vectorised the input texts using Term Frequency-Inverse Document Frequency by adding a minimum threshold of 5 times in at least half of the total number documents. He also used the prompt questions and the proficiency levels as input features, and achieved around 83.6% accuracy. A similar set of features, also for an SVM, was used by Jarvis, Bestgen, and Pepper (2013),

[3] In the `scikit-learn` Python package, which we are using for a part of our study (see more in Section 3.3.1), the function words that its vectorisers use for extracting features are assigned to a `frozenset`, i.e., an immutable collection of items. By diving into `scikit-learn`'s (Pedregosa et al. 2011) source code, we find the frozenset for the English stop words (at the time of writing only English stop words are supported). The list is provided by the Glasgow Information Retrieval Group `http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words`. For example, according to Stamatatos (2009, 540-1), many different sets of function words have been used for English. See the reference to the respective file on the latest commit: `https://github.com/scikit-learn/scikit-learn/blob/main/sklearn/feature_extraction/_stop_words.py`.

who secured the first place in the 2013 NLI Task by scoring 83.6%. The difference is that they used word, character and POS tags n-grams of $n = (1, 3)$ present in at least 2 texts of the TOEFL11 corpus, while also adding log-entropy weighting. Finally, they extended their feature set by including lemmas, namely the dictionary entries of the lexemes present in the texts. The exact opposite of such a complex set of features is what Kulmizev et al. (2017) employed, albeit for the second iteration of the task (2017). They achieved an accuracy of 87.6% by training a linear SVM with character-only n-grams of $n = (1, 9)$. This proves that character n-grams were the top single-type features (Malmasi et al. 2017, 70). The dominance of the SVMs in the NLI field and the two Shared Tasks in particular although evident, benefited from the inclusion of a second classifier in the case of Cimino and Dell'Orletta (2017). The team, that achieved the second highest accuracy and F1 scores in the 2017 Shared Task, used a stacking approach based on two classifiers: a Logistic Regression sentence-level one and an SVM one for the documents, although the results from the SVM alone were already high enough. They used lexical and syntactical features to secure the top spot on the leader board. More concretely, they used text and average word lengths, along with character, function word, word, and lemma n-grams. They applied smoothing on their character features, binary representation of their function words and words features, and normalised relative frequency of the lemmas. As far as their syntactic features are concerned, they used coarse-grained POS n-grams, simple and combined with lemma n-grams, linear and hierarchical dependency types n-grams, and head-dependants syntax tree. Another novel approach, at least in the context of NLI research up until 2017, was followed by Markov et al. (2017). The authors, along with the commonly-used features found in related research (word, lemma, character, and function word n-grams), added character n-grams from misspelled words, 10 categories of character n-gram features, and syntactic features using the Stanford dependency parser and an algorithm provided by Posadas Durán et al. (2017), primarily used for authorship attribution. Their liblinear SVM (with 'OVR' multi-class strategy), equipped with log-entropy weighting, achieved the second-highest score (88.1%) in the 2017 Shared Task, but shared the first rank since there was only a single class difference with Jarvis, Bestgen, and Pepper (2013). They pre-processed the corpus by replacing digits with zeros and also including skip n-grams. The researchers conclude that the addition of spelling-error character n-grams does improve the accuracy of the model, which is inline with Koppel, Schler, and Zigdon (2005b). Their approach into implementing them follows categorising character n-gram features according to Sapkota et al. (2015) to extract differences in the sub-word level, e.g., the two occurrences of *less* in the phrase *less carelessness* (Markov et al. 2017, 376).

To sum up, regarding the 2013 Shared Task, 13 out of the 24 participating teams trained an SVM model (mostly with the liblinear solver), and all but one of the 5 teams with the highest scores used n-grams of $n \geq 4$ for characters, POS and word n-grams, along with syntactical dependency features (Tetreault, Blanchard, and Cahill 2013, 54-55). For the Shared Task's second iteration (Malmasi et al. 2017), deep learning approaches did not outperform traditional ML approaches (i.e., SVM classifiers). It is important to clarify that the 2017 NLI Shared Task included three tracks; written-only, spoken-only, and combined, based on the type of data used. Malmasi et al. (2017) report that combining the standard written texts with spoken counterparts benefit the classification performance. However, we focused on the first track only, since the other two do not fall inside the scope of this paper. Last but not least, it is interesting that only few of the top-scoring teams (and even fewer for the rest of the participating ones) in

both years used document-based features, e.g., (average) word, sentence and document lengths.

To conclude, the two main studies that used the L2-Reddit corpus after its release are Goldin (2019) and Goldin, Rabinovich, and Wintner (2018). In spite of the objective of the latter being to study the influence of cognates in L2 English, their findings suggest that NLI is possible on social media posts and comments. In addition, Goldin (2019) trained a Logistic Regression classifier including all the previously used features and achieved 81% when evaluated in-domain, and 72% out-of-domain. All in all, we see that there were many different approaches found in the literature for NLI. This gives us a plethora of features to experiment with, which we will discuss in detail in Section 3.3.1.

2.2.2 Neural networks

Compared to the previous section about the work that has already been done by implementing ML models, there are a lot fewer DL approaches followed to tackle the NLI research. However, as we will see throughout this section, the approaches that researchers followed in their NNs implementations vary substantially.

Bjerva et al. (2017) focused on applying neural networks on the NLI task. In fact, they experimented with including and excluding external resources (features) from their Deep Residual Networks (ResNets), which are a class of CNNs that utilise a building block called a "residual block" and allows the network to skip or bypass certain layers of the network, making it less prone to the vanishing gradients problem (He et al. 2015).

Chan et al. (2017) compared the performance of two ensembles, one comprised SVM classifiers and the other Fully Connected Neural Networks, on the 2017 Shared Task dataset. The SVM ensemble outperformed the FCNNs when all features were used in training (87.3% compared to 85.6%) on the test set. It is, however, noteworthy that the FCNN ensemble performed better on 5 out of 7 feature sets (excluding the POS and the phoneme ones).

Goldin (2019), apart from their Logistic Regression approach mentioned in the previous section (Section 2.2.1), implemented a Long Short-Term Memory (LSTM) network with Attention using Keras[4] and the Adam optimiser, and a classification layer consisting of 30 neurons, one for each L1 present in their subset of the L2-Reddit corpus. Their model, in its simplest configuration (with $V = 5,000$), achieved 73.6% in the in-domain and 51% in the out-of-domain test sets. After experimenting with different variations, their best model included spelling and auxiliary output, and resulted in increasing the accuracy by a significant amount, albeit only in the in-domain dataset (77.2%). A similar approach of comparing the two types (namely statistical and neural) of models was followed by Goldin (2019).

Oh et al. (2017) compared two Deep Neural Networks (DNNs): (a) a vanilla DNN-based ensemble classifier for late fusion; and (b) a multi-column deep-stacking DNN-based one for early fusion, and concluded that the latter performed better when there were significant differences in terms of the performance of the heterogeneous features. Their latter classifier achieved 86% for the essay-only track of the 2017 NLI Shared Task, making them the second participating team who employed Deep Learning and secured a relatively high place on the leader board (Malmasi et al. 2017).

[4] https://keras.io.

Last but definitely not least, Lotfi, Markov, and Daelemans (2020) exploited OpenAI's Generative Pre-trained Transformer-2 (GPT-2) (Radford et al. 2019), fine-tuned it on texts written by authors with the same L1, and used the model to classify an unseen text by determining the minimum language model (LM) loss compared to the fine-tuned GPT-2 model. Their generative approach outperformed the previous state-of-the-art (88% by Markov et al. 2017) with an accuracy of 89% on the TOEFL11 test set. They also achieved excellent results, 86.8% and 94.2% accuracies, on the TOEFL11 and the ICLE corpus and with 10-fold and 5-fold CV, accordingly. Their approach introduces the true potential of generative models in the NLI task, and they hypothesise that it can be used for other style-based classification tasks, given that the samples, at least in the test data set, are long enough to ensure a significant LM loss.

We can clearly see that as far as DL approaches are concerned, there has been a plethora of different networks built to tackle the NLI task. The approach we followed as part of this paper will be explained in detail in Section 3.3.2.

3 Methodology

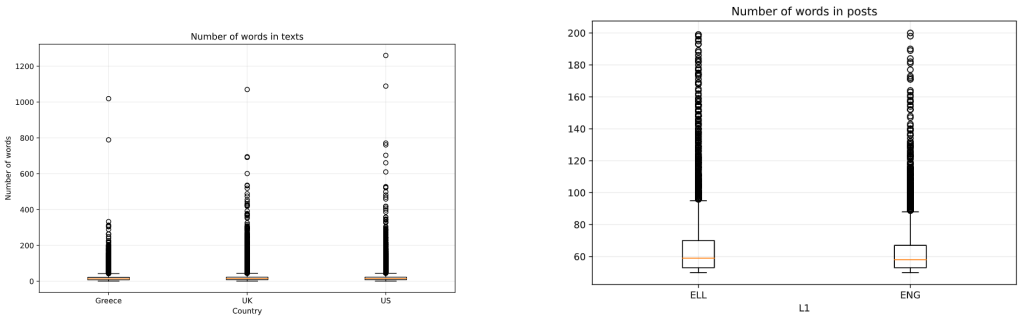
As shown in Section 2.2, all previous research treats the NLI task as a classification problem. This is to be expected, as we try to predict the class label of the native language of each author. In the previous section, it was also made clear that SVM is the preferable algorithm when applying traditional Machine Learning approaches, along with some studies reporting achieving higher results with Logistic Regression. This is the main reason behind selecting both of them as classifiers for the task at hand. For the Deep Learning approach, a BERT-based and a DistilBERT model have been chosen for fine-tuning.

3.1 Data

In order to tackle the task of NLI, it would be ideal to use a corpus which contains English texts written by Greek and English L1 authors respectively given as similar a set prompts as possible. As we have seen in Section 2.1, topic bias, primarily caused by the models laser-focusing on content words of the input texts, can be a challenging issue to overcome. Therefore, by minimising the range of topics the texts have been written about, we can have a clearer view on what the model *actually* learns during the training phase. A possible candidate would be the recently-released, third version of the International Corpus of Learner English (ICLEv3) corpus (Granger et al. 2020). Therein, 462 essays, accounting for a total of approximately 265,000 words from the Aristotle University of Thessaloniki, are included which were written by L1 Greek university students. Unfortunately, the ICLE corpus does not contain essays written by English native speakers, and thus the requirement for the texts to be as similar as possible to facilitate the binary nature of the problem we are tackling is not met. Instead, we will shift our focus from texts written by learners to social media posts, and use a subset of the L2-Reddit corpus (Rabinovich, Tsvetkov, and Wintner 2018, introduced in Section 2.1.3). A brief summary on the corpus is given in Section 2.1.3. According to the creators of the corpus, it "contains spontaneous productions of advanced, highly proficient non-native speakers", except, of course, when referring to English L1 users (Rabinovich, Tsvetkov, and Wintner 2018, 331).

3.1.1 Subset extraction

We extracted the corpus and the files containing the raw texts per country. Since we are interested only in English and Greek L1 users, we filter out only the files containing posts uploaded by users from Greece, the UK, and the US which we will combine later in a single file for L1 English (see Section 3.1.2). The length of the texts varied from 1 up to 3,000 words, as shown in Figure 2a. In line with previous research (e.g. Goldin 2019), and given the distribution of the word lengths in the two L1s, we filtered out the texts with length less than 50. We then sampled the same number of texts from the two English-speaking countries as in the Greek subset to make the two datasets as balanced as possible. Finally, we generated three distinct subsets of the corpus: one consisting of texts shorter than 200 words, one with texts shorter than 300 words, and a third with no length restriction. We ran the experiments on all 3 of the different subsets to examine whether the maximum length (`maxlen`, ℓ_{max}) affects the performance of the models. Table 1 presents a comparison of the L2-Reddit corpus and the subset utilised in our experiments with the ICLEv2 and TOEFL11 datasets.



(a) **(b)**

Figure 2: Boxplots with the number of words (a) by specified country of authors; and (b) by language, after sampling the same number of texts as in the Greek (`ELL`) subset from the UK and the US subsets for English (`ENG`). The orange lines denote medians.

The L2-Reddit corpus is available as a collection of files, one for each country of origin of the authors of the posts and comments on said social platform. Since Greek is the official language only in Greece[5], we assign a new column on our data containing the L1 of the authors present in our samples. We then proceed to do the same for the UK and US subsets. To comply with the TOEFL11 corpus, we use the 3-letter language/locale codes to represent each L1, as standardised in the second part of ISO 639-2[6]. Henceforth, Greek is denoted as `ELL` and English as `ENG` when referring to the class labels.

The corpus is available in many variations[7], one of which includes the tokenised texts in a single CSV file for each country. We used three CSV files, which contain the texts written by authors with Greek, UK, and US origin, respectively. The first

[5] Greek is also an official language in Cyprus, but (a) the Cypriot dialect differs significantly (see, for example, Newton 1972), and Cyprus used to be part of the British Commonwealth until mid-20th century. We thus do not take the corresponding file into account.

[6] See full list of codes at `https://www.loc.gov/standards/iso639-2/php/code_list.php`.

[7] `https://drive.google.com/drive/folders/125RAHvCIHBR-jAUnIhqzWhdxh0mQ_fcv`.

Corpus	L1s	N texts	Unique topics
ICLEv2	7	6,085	736
TOEFL11	11	11,000	8
L2-Reddit	46	>15,000,000†	>80,000*†
L2-Reddit (here)			
$\ell_{max} = 200$	2	9,888	1*
$\ell_{max} = 300$	2	9,912	1*
$\ell_{max} =$ none	2	9,924	1*

* Here, we treat subreddits as topics, as they operate as independent forums, each of which addresses a single general topic of discussion (e.g. `r/Europe`, `r/AskEurope`).

† This number accounts not only for the main corpus, which contains texts posted in 5 Europe-related subreddits by users from 31 countries, but also for texts that they posted in other, unrelated subreddits (Rabinovich, Tsvetkov, and Wintner 2018).

Table 1: Comparison of the basic characteristics between the corpora used for NLI in the past and the one used in this study.

two columns of the file are filled in with the author's username and the subreddit they posted the text in. Following the conversion of the CSV files into three distinct `Pandas` DataFrames each one consisted of three columns (`username`, `subreddit`, and `text`), we anonymised the usernames by replacing them with a randomly-generated hexadecimal alphanumeric sequence. This will be proven to be useful later, during train-test splitting. Then, we appended a column at the end of the DataFrames containing the class label, namely the L1 of the text authors, with `ELL` denoting Greek and `ENG` denoting English in the corresponding datasets, as previously mentioned. Finally, we calculated the total number of the `ELL` texts and sampled an equal amount of `ENG` texts, the first half of which originated from the UK and the other half from the US subsets, accordingly, to represent the *lingua franca* as much as possible. This was done to ensure that the two classes are equally represented in the samples and, as a result, that the data is perfectly balanced. As explained in Section 3.1 earlier, we finally produced three versions of the final subset, one including all the texts, and the other two by setting a maximum word length threshold of 200 and 300 words, respectively. It is however evident that the two latter versions are not perfectly balanced, and it remains to be seen whether this will influence the final results of the models.

3.1.2 Pre-processing

As mentioned above, we used the already tokenised version of the L2-Reddit corpus, which separated the punctuation marks from the words they were initially attached to and also split contracted forms into separate tokens (e.g., *shouldn't* → *should n't*). Previous studies have reported on aggressive text cleaning having negative impact on performance (Boumparis, den Ouden, and Gevers 2022; Boumparis 2022), which resulted in us only replacing digits with zeros, a pre-processing step that has been done before by Markov et al. (2017), as part of the 2017 NLI Shared Task model development, and by Markov, Stamatatos, and Sidorov (2018). This ensures that the format which

large (e.g., $00,000$ or 00.000) and floating numbers (0.00 or $0,00$) are written in is captured while, at the same time, the numeric values themselves are not.

3.2 Feature Engineering

For the statistical algorithms, various combinations or individual features and two main feature sets were tested prior to achieving the highest results. Below, we will analyse the ones that contributed the most, after we first discuss the different configurations that were explored. Both classifiers were trained using `sklearn`'s SGDClassifier[8]. Here, we fit and compare Linear Support Vector Machines and Logistic Regression classifiers.

3.2.1 Linguistic features

The first feature set comprises different representations of the text on the character and word levels by using n-grams. N-grams are consecutive sequences of n items – in our case, words, characters, or Part-of-Speech tags. The use of n-grams as features in NLP tasks is motivated by their ability to capture local syntactic and semantic patterns within a text (Brown et al. 1992). By considering the context provided by adjacent words or characters, n-grams can provide valuable information about the meaning and structure of a piece of text. In addition, they are relatively simple and efficient to compute, making them a popular choice for feature extraction. In the following subsection, we will delve into the specific details of how we extracted and employed word, character, and POS n-grams for our statistical models. The impact of different n-gram lengths and the trade-offs involved in their use will be discussed later in Section 4.

Word *n*-grams The most basic representation for any text is vectorising the words that it is consisted of. We experimented with two main configurations for dealing with words. First, we ran our two statistical algorithms with individual n-grams with ranges $\{(n, n) \mid n \in \{1, ..., 5\}\}$, and then including the individual ranges, namely $\{(1, n) \mid n \in \{1, ..., 5\}\}$. Word unigrams were included, even though they make the models prone to overfitting due to topic bias, as there are certain words that are used significantly more frequently in the texts that belong to each one of the classes. In the scatter plot found in Figure 3, a visual representation of all the unigrams is shown[9], and the words whose usage frequency is significantly different between the two L1s in question are denoted. For example, it is evident – and expected – that words such as *Greek(s)* and *debt* as well as *UK* and *British* are more likely to be found in texts written by L1 Greek and English authors, respectively. It is interesting, nonetheless, that we see a significant difference in how the authors of these two L1s are using some function words, e.g., *it* and, more interestingly, the difference in *will* and *was*, on the Greek half of the graph, and, conversely, *would* and *be* on the English half.

Character *n*-grams As mentioned in Section 2, character n-grams were the single most contributing feature in the 2017 NLI Shared Task (Kulmizev et al. 2017; Malmasi et al. 2017, 70), and many teams employed $n \geq 4$. We followed a similar approach on the

[8] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html.

[9] Most overlapping data points in the interval $x, y \in \{0, 20\}$ that are not significantly different were removed from the vector graphic to reduce its file size (from $\simeq$ 2 MB to $\simeq$ 250 kB) in almost lossless fidelity.

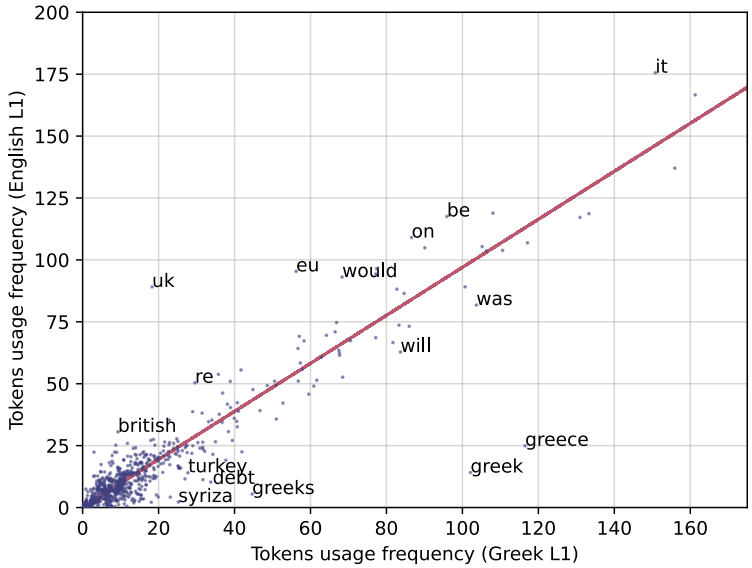


Figure 3: Scatter plot showing the usage frequency of word unigrams in the corpus between L1 English and Greek authors, respectively. The words displayed have a difference of at least 20 between the two groups.

implementation of these features to the word n-grams that we described earlier, i.e. single and sequential ranges of (n, n) and $(1, n)$, respectively. What is, however, different is that, in the case of the character n-grams, we experimented with binary representation as well. Albeit less informative than Term Frequency-Inverse Document Frequency, binary representing n-grams as present (`1`) or absent (`0`), on the other hand, decreases the possibility of overly focusing on capturing the full context which could potentially lead to the overfitting of the model, given that the prompts are not identical nor as strictly similar among the authors of the Reddit posts as, for example, the prompts found in the TOEFL11 corpus.

Universal Part-of-Speech *n*-grams Function words are lexical items that serve to clarify the relationships between the content-bearing elements of a sentence and facilitate the construction of syntactic structures such as verbal complements, relative clauses, and questions (Smith and Witten 1993). We added a new column that contains the tokens in each text but replaced by the Universal Part of Speech (UPOS) to which they belong. To do this, we used the `spaCy` library[10] which uses the Universal Dependencies tagging algorithm under the hood which, in turn, is based on the (universal) Stanford dependencies parser (de Marneffe et al. 2014), the Google universal Part-of-Speech tags (Petrov, Das, and McDonald 2011), and the Interset interlingua for morphosyntactic tagsets (Zeman 2008), and contains a closed set of 17 POS tags in total[11]. In addition to replacing each token with its corresponding POS tag, we also followed the approach

[10] https://spacy.io/usage/linguistic-features#pos-tagging.
[11] https://universaldependencies.org/u/pos/.

introduced by Markov, Nastase, and Strapparava (2022, 173) according to which only the content words are being replaced by the POS tag, and each function word is left as is. Table 2 shows an example of a tokenised sentence and its additional representations using UPOS unigrams and function word (FW) features.

Tokenised sentence	*I do n't exactly disagree with this notion.*
UPOS unigrams	PRON AUX PART ADV VERB ADP DET NOUN PUNCT
UPOS unigrams & FW features	I do n't ADV VERB with this NOUN .

Table 2: A sample tokenised sentence from the corpus and how it was represented using only Universal Part-of-Speech (UPOS) unigrams and then a combination of the tags with function word (FW) features.

Misspelled words Spelling errors are another feature that we saw having been extracted regularly in other NLI studies in Section 2. For example, Koppel, Schler, and Zigdon (2005a) extracted the spelling errors using Microsoft Word. Our extraction method was done in a different way, however, which is more in line with Kulmizev et al. (2017). In fact, we used the `PyEnchant` library[12] to extract the misspelled words by creating a custom Python `class` which inherits from `sklearn`'s `TfidfVectorizer`. Creating a class tailored to our use case was necessary since (a) the corpus contains both British and American English, and thus there are words that are spelled differently between the two dialects (e.g., *defence* and *defense*, *travelled* and *traveled*); and (b) there are many acronyms, proper nouns and names of places, political parties, and persons of interest (e.g., *Nafplion*, *PASOK*, *Giannis Antetokounmpo*) that are significantly more likely to reveal the L1 of the author since they are mainly present in the L1 Greek subset. First, our `MisspellingVectorizer` checks whether the word meets the requirements to be spellchecked, i.e., it contains only Unicode letters[13], is at least 3 characters long, and not a proper noun or an acronym (by naïvely checking whether the first letter is capitalised). Then, if the word should be checked, it is looked up into both the British and the American English dictionaries provided by `PyEnchant`. The word is, finally, added as a feature only in the case that it is not found in *any* of the two dictionaries. From the 1,633 texts in the testing dataset, a total of 596 words were extracted and included as features in this step. Lastly, we once again experimented with binary or frequency-based representing these word spelling errors.

Punctuation marks In NLP, and even more so in stylometric experiments (see, for example, Stamatatos 2009), punctuation marks are a robust feature that can meaningfully contribute to a model's ability to predict the target variable ($\hat{y}$). Especially in the case of NLI, numerous previous studies (for reference, see Koppel, Schler, and Zigdon 2005a; Wong and Dras 2011; Malmasi and Dras 2017; Markov, Nastase, and Strapparava 2018) have shown that patterns in punctuation usage is a primary example of cross-linguistic influence, namely that the author's L1 affects the use of certain linguistic elements in L2 texts. However, others have shown that removing them does not result in significant difference of the classification performance (Wu et al. 2013). We opted to represent

[12] `http://pyenchant.github.io/pyenchant/`.
[13] `https://docs.python.org/3/library/stdtypes.html#str.isalpha`.

the instances of punctuation marks by simply running each token against the list of punctuation marks found in Python's `string` standard library module, extracting their relative frequency, and then normalising it around the mean ($\bar{\mu}$).

3.2.2 Pre-trained embeddings

Finally, we opted to use pre-trained word embeddings to save time and resources, as it is more efficient to fine-tune a pre-trained model rather than training a new one from scratch.

In NLP, word embeddings are used to represent the meanings of words in a continuous, low-dimensional vector space, and are found to achieve very high scores, and even state-of-the-art accuracies in text classification[14]. This allows machine learning models to capture the relationships between words more easily. There are various methods for learning word embeddings, such as word2vec (Mikolov et al. 2013) and GloVe (Pennington, Socher, and Manning 2014). Although robust, none of these two models are up to date, and GloVe, despite being more performant than word2vec, does not make use of neural networks in the background. Another option for obtaining pre-trained word embeddings is to use those provided by OpenAI[15]. These embeddings have a number of advantages compared to the ones listed earlier. In particular, they have been trained on an even larger dataset which dates fairly recently (September 2021) and were released in mid-December 2022, allowing them to capture the relationships between a wider range of words and concepts. Additionally, they have a substantially larger maximum output compared to their first-generation counterparts (from 2,048 to 8,192), even though our texts are nowhere close these limits in terms of length. For best results, according to OpenAI's documentation[15], the cosine similarity (S_C) was used.

The pre-trained model produced embeddings of magnitude 1,536 for each text, regardless of the latter's length in words. We then expanded it into the equal number of columns so that a single floating point number is placed in each cell, prior to feeding it to our classifiers.

3.3 Experimental Setup

Our approach in the experiments we conducted is two-fold: (a) we study the difference in performance between linguistic features, on the one hand, and pre-trained large language model and word embeddings, on the other; and (b) we examine whether combining the 3 individual classifiers in each case into an ensemble setup with hard majority voting produces higher results than the individual classifiers that do not rely on linguistic features. In others words, we want to evaluate the contribution of linguistic features, given the praise they get though the course of time since NLI became an NLP task, as we have already discussed in Section 2.

3.3.1 Statistical classifiers

It is now a good opportunity to mention that the hyperparameter tuning took place early on in the experimentation with 10-fold Cross-Validation (CV) using Grid Search and resulted in the default ℓ_2 regularisation as the penalty that yields the best per-

[14] https://openai.com/blog/new-and-improved-embedding-model/.
[15] https://beta.openai.com/docs/guides/embeddings.

forming models, along with an extensive search with $C = \{10^k \mid k \in \{-3, ..., 3\}\}$ for the Linear SVM and $C = \{10^k \mid k \in \{-4, ..., 2\}\}$ for the Logistic Regression classifier. For these parameters, $C = 1$ and $C = 10^{-4}$ were selected as the optimal values, respectively. A table listing which hyperparameters were selected as the best ones is shown below (Table 3).

			Characters	*n*-gram range		
Model	α	**Penalty**	**min *df***	**POS tags**	**Words**	**F1-score**
LSVM	0.001	ℓ_1	0.1	$n = (1, 1)$	$n = (1, 2)$	0.773
	0.01	ℓ_2	0.25	$n = (1, 2)$	$n = (1, 2)$	0.770
	0.01	ℓ_2	0.1	$n = (1, 2)$	$n = (1, 1)$	0.768
	0.01	ℓ_1	0.1	$n = (1, 1)$	$n = (1, 1)$	0.768
	0.001	ℓ_1	0.1	$n = (1, 2)$	$n = (1, 1)$	0.764
	0.0001	ℓ_1	0.25	$n = (1, 2)$	$n = (1, 3)$	0.763
	0.1	ℓ_2	0.1	$n = (1, 3)$	$n = (1, 3)$	0.763
	0.001	ℓ_1	0.25	$n = (1, 1)$	$n = (1, 2)$	0.762
	0.0001	ℓ_1	0.1	$n = (1, 1)$	$n = (1, 3)$	0.762
	0.1	ℓ_2	0.25	$n = (1, 3)$	$n = (1, 3)$	0.761
LogReg	0.01	ℓ_2	0.1	$n = (1, 2)$	$n = (1, 3)$	0.771
	0.001	ℓ_2	0.25	$n = (1, 2)$	$n = (1, 3)$	0.771
	0.0001	ℓ_2	0.25	$n = (1, 2)$	$n = (1, 2)$	0.771
	0.1	ℓ_1	0.1	$n = (1, 2)$	$n = (1, 3)$	0.769
	0.1	ℓ_1	0.1	$n = (1, 1)$	$n = (1, 2)$	0.767
	1	ℓ_2	0.25	$n = (1, 1)$	$n = (1, 1)$	0.765
	0.1	ℓ_2	0.25	$n = (1, 2)$	$n = (1, 1)$	0.765
	0.001	ℓ_2	0.1	$n = (1, 2)$	$n = (1, 2)$	0.764
	0.001	ℓ_1	0.25	$n = (1, 3)$	$n = (1, 2)$	0.764
	0.0001	ℓ_2	0.25	$n = (1, 2)$	$n = (1, 1)$	0.763

Table 3: Top 10 best hyperparameters for each of the two linear classifiers.

Baselines Since the texts in our corpus are equally distributed between the two target classes, a random baseline is automatically set at 50%. We ran a simple Bag-of-Words (BOW) model the highest thereof achieved 78.5%. During all our train and test splits, we set aside all the texts by the same author ID during each fold. This was crucial to ensure that no leakage would take place between the two datasets and result in overfitting of the models.

Stochastic Gradient Descent Binary classification is a problem that linear classifiers can tackle with great reliability. They are able to fit the best possible hyperplane that separates the two classes to which the data belong. These classifiers generally produce even higher results when combined with the SGD optimisation algorithm that updates the values of the coefficients and biases with the goal of minimising the loss. In `scikit-learn`, it is very straight-forward to implement SGD into (linear) classifiers,

by setting the desired loss function (hinge or squared hinged for Linear SVM, and log[16] for Logistic Regression).

10-fold Grid Search Cross-Validation However, `loss` is not the only parameter that needs to be set. The main other ones are related to the hyperparameters of the model, which the `scikit-learn` package calls α, even though it refers to the C parameter in the case of individual classifier instances, namely without the inclusion of the SGD optimisation. For the case of the Linear SVM classifier, the 10-fold Grid Search Cross-Validation that we ran in all cases, selected $C = 1$ as the best-performing value, whereas for the LogReg classifier, a C equal to 10^{-4} was preferred. In addition, the penalty parameter was also added to the grid, with ℓ_1 and ℓ_2 as possible candidates. However, the results were mixed, as neither of the two was clearly yielding higher results. In the case of the 10 Linear SVM classifiers with the highest results, ℓ_1 was selected 6 times, conversely to ℓ_2 that is present 4 times in the best parameters `dict`. The opposite is true for the Logistic Regression model, where the latter is present in 6 runs as the penalty of choice. It is important to point out that none of these results were not significantly higher, either, ranging from 76.3% to 77.3%.

During Cross-Validation, all the different features we discussed in Section 3.2 were tested, after the above hyperparameters were selected. One final parameter that was given special attention to is the minimum document frequency (min_*df*). A few very low values were originally selected for hyperparameter tuning, but no significant correlation between these values and the results could be extracted. Again, similar to the penalties described in the previous paragraph, the preference given via Grid Search is somewhat split, leading to no definitive choice of any value as optimal or even contributing to better accuracy. Table 3 shows how much the different hyperparameters contibuted to the mean F1-macro scores during CV, where it is obvious that min_*df* does not play any significant role in performance and score, as its different values are present in both higher and lower results – indicating F1-macro scores – withing the $\approx$ 75-77% range. This had as a result us not using this parameter in our experiments thereafter.

3.3.2 Neural classifiers

HuggingFace Transformers As far as the neural branch of our experiments is concerned, in spite of our approach being a lot more direct, it required as much experimentation for the models as in the case of the statistical ones. We employed two large language models, $\text{BERT}_{\text{uncased}}$ and $\text{DistilBERT}_{\text{uncased}}$ respectively, for text classification using the `Transformers` library by HuggingFace[17]. First, we split the data into 3 subsets (80% train, 10% validation, and 10% test), and again took all necessary precautions to avoid leakage, by ensuring that no text from the same author is present in more than one of the datasets. Given the small percentages of the development and test sets, the total number of samples in there varies slightly. We enabled automatic truncation for texts longer than 512 tokens (default value for the two language models) along with padding for the input texts that were shorter than this default value. We experimented with different values for the model's learning rate ($5\text{e}{-5}$, $2\text{e}{-5}$, $1\text{e}{-6}$, and $1\text{e}{-7}$). The highest results without signs of overfitting by plotting the losses were

[16] Note: Since version 1.1, the `loss` parameter that would fit a LogReg model, if set to `'log'`, was deprecated and will be removed in an upcoming version. We used the future-proof value in the code for model development.

[17] https://github.com/huggingface/transformers.

achieved with $1e{-}7$. Our models were fine-tuned using 6 layers, an attention dropout of 0.1 to randomly drop out elements from the softmax in the attention equation, 768 dimensions and $3,072$ hidden dimensions for the feature vectors, a dropout of 0.75, and the GeLU activation function to combine the dropout regularisation with what a ReLU activation function does. The total number of the vocabulary (V) is $30,522$. The best model was trained on a total of 37 epochs by implementing a patience of 3 epochs for early stopping. The experiments took place only on the texts with no maximum length limit (ℓ_{max}) to save on time and resources. This was also the reason behind training a DistilBERT model since its creators claim that they "reduced the size of a BERT model by 40%, while retaining 97% of its language understanding capabilities and being 60% faster". For the batch size, we started with 16 and then increased it to 32, which yielded better results along with smoother training and validation loss curves. This was also the optimal value for the best model. Finally, for optimisation, we combined the Adam optimisation algorithm with the weight decay regularisation method (`AdamW`), resulting in faster convergence, which meant that it did not require further tuning. Our results suggest no presence of overfitting (see Section 4.1.2 for details).

3.3.3 Ensemble classifier

Last but not least, a combination of individual classifiers into a voting ensemble was attempted. The ensemble consisted of three linear classifiers, all equipped with SGD, but with a different set of features. Into the first two models, a Linear SVM and a Logistic Regression, all the linguistic features we listed previously in Section 3.2.1 were fed. The third component was another `SGDClassifier` whose loss function was selected after 10-fold Grid Search Cross-Validation (a Logistic Regression classifier was selected as optimal post hyperparameter tuning). Only the word embeddings representations were given to the latter pipeline. Then, hard voting was implemented to let the majority of the predicted target values to be assigned their definitive label based on the predicted value of at least two of the ensemble's components. All results are shown and analysed later in Section 4.2.

4 Results and Discussion

Following the methodology analysed in the previous main section and after experimenting with various combinations of feature sets, features, algorithms, and hyperparameters, it is now time to present the results and discuss about them.

4.1 Individual Classifiers

Evidently, a lot more time and effort went into developing robust statistical pipelines, however all classifiers apart from one configuration surpassed both baselines (Figure 4).

4.1.1 Statistical algorithms

All statistical models, apart from the baselines and the two default configurations with word unigrams and simple word embeddings, were implemented using custom scikit-learn transformers (to be later fed into its `Pipelines`) on the data and required many attempts to finally yield the best-performing model. The highest results obtained from each combination of statistical algorithms with the various feature sets and features are

presented in Table 4, with the overall highest scores being denoted in bold. From the first glance, the majority of the results are very close with one another and, especially in the linguistic feature set, there is a tendency for the simplest amount of features to yield higher results. In line with Kulmizev et al. (2017), simple character n-grams with $\{n, n \mid n \in \{1, ..., 9\}\}$ achieved the highest mean in terms of F1-macro score (79.7%). It is also worth noting that both the Linear SVM and the Logistic Regression models performed very similar across the board – although one would outperform the other consistently in some cases – making it impossible to select only one for the rest of the experiments. Detailed tables listing all results can be found in the Appendix.

	Linear SVM		**LogReg**	
Features	mean ($\bar{\mu}$)	SD (σ)	mean ($\bar{\mu}$)	SD (σ)
Baselines				
Random n-grams	0.500000	–	–	–
Bag-of-Words (BOW)	0.745961	–	–	–
***n*-grams**				
Word unigrams	0.774693	0.053506	0.783484	0.052771
Individual $(\{n, n \mid n \in \{1, ..., 9\}\})$	**0.796610**	0.045355	**0.797145**	0.055094
+ Min. *df*	0.772540	0.151395	0.782999	0.153387
+ Misspellings	0.783624	0.045767	0.786697	0.042822
+ Misspellings + FW	0.701601	0.027629	0.717334	0.023856
Continuous $(\{1, n \mid n \in \{1, ..., 10\}\})$	0.771822	0.128783	0.780557	0.178545
Word embeddings				
Simple	0.817780	0.043755	0.819092	0.038526
+ Character n-grams	**0.825552**	0.048806	**0.826143**	0.049665
+ UPOS n-grams	0.818601	0.048108	0.815482	0.045965
+ UPOS n-grams (bin.)	0.810953	0.041305	0.819494	0.038318
+ UPOS + Character n-grams (bin.)	0.820989	0.045070	0.822507	0.044193

Table 4: The highest mean of F1-macro scores following 10-fold Cross-Validation per feature set along with the two baselines between Linear SVM and LogReg. The overall highest scores per feature set are denoted in bold.

If we dive deeper, however, and visualise the different components of the linguistic feature set, it is very interesting to see how these performed per ℓ_{max}. A group of such plots is available on Figure 5 below. First, the line plots confirm our hypothesis that the number of total words does not affect the performance significantly. However, as we will see in Section 4.4, this is true only when approaching total length in words from the right, namely its maximum possible values. Second, we see that the more abstract the representation of the words is, the more robust and higher the results. Character and POS n-grams show little to no difference in F1-macro scores regardless of their range, conversely to words whose performance drops the larger the range of n is. The exact numeric values of the mean F1-macro scores are available in table form in the Appendix.

The combination of linguistic features generally performed well in the NLI task at hand, but the pipelines that employed word embeddings consistently outperformed other feature combinations. In fact, they did not score below 80%, even in their simplest

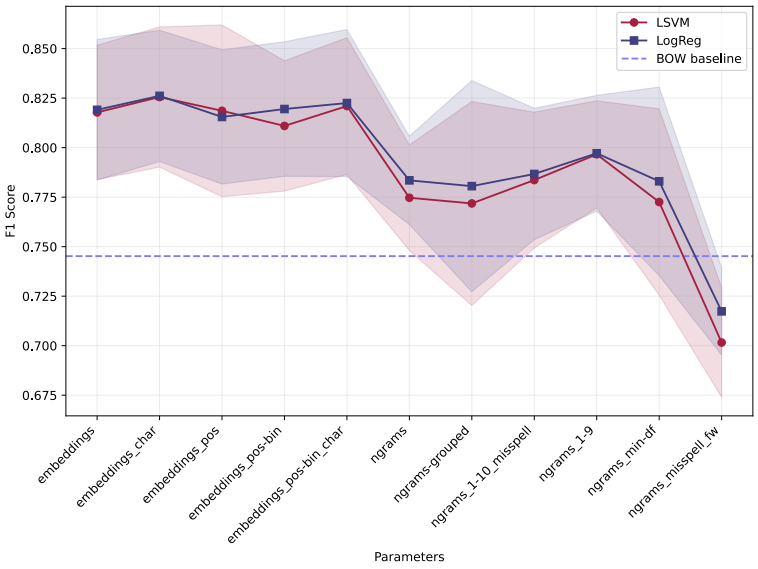


Figure 4: Comparison of the highest in average F1-macro scores following 10-fold CV with the two individual statistical classifiers, Linear SVM and Logistic Regression, per feature set. The horizontal line denotes the BOW baseline.

form ($\approx 81.8\%$), namely without adding extra features on the texts. This goes to show that fine-tuning on pre-trained embeddings can not only be proven a higher-scoring choice but it is also crystallised on the mean fit time parameter. Word embeddings training took approximately 2 hours on the same machine that the neural classifiers were train on, but every pipeline that involved them took significantly less time compared to linguistic-only feature extraction and selection. This suggests that the use of word embeddings can not only improve the performance of text classification models but also make the training process more efficient and decrease its carbon footprint.

4.1.2 Neural algorithms

As far as the algorithms that were based on pre-trained LLMs are concerned, the results are very promising, despite the fact that we cannot be sure how the models achieved these scores. Getting quite high results like these is a solid proof of the reliability and robustness of BERT-based models for classifying text. Even though there is a tenable, we think, hypothesis that the BERT's 'style' is close to average, since the individual stylometric fingerprints on its training data are suppressed given the large amounts thereof, it manages to identify (at least some, if not most, of) the patterns that differentiate ELL from ENG. The high scores of 82.6% and 83.1% and the absence of overfitting as indicated by the loss curves (shown in Figure 6) further support the reliability of BERT-based models for this task. One satisfactory explanation for the effectiveness of BERT-based models in this task is their ability to capture contextual information through their training process, which involves masking certain words in the input and predicting their identity. This understanding of language context seems to be particularly useful in distinguishing between ELL and ENG text, as the texts that belong to each one are written

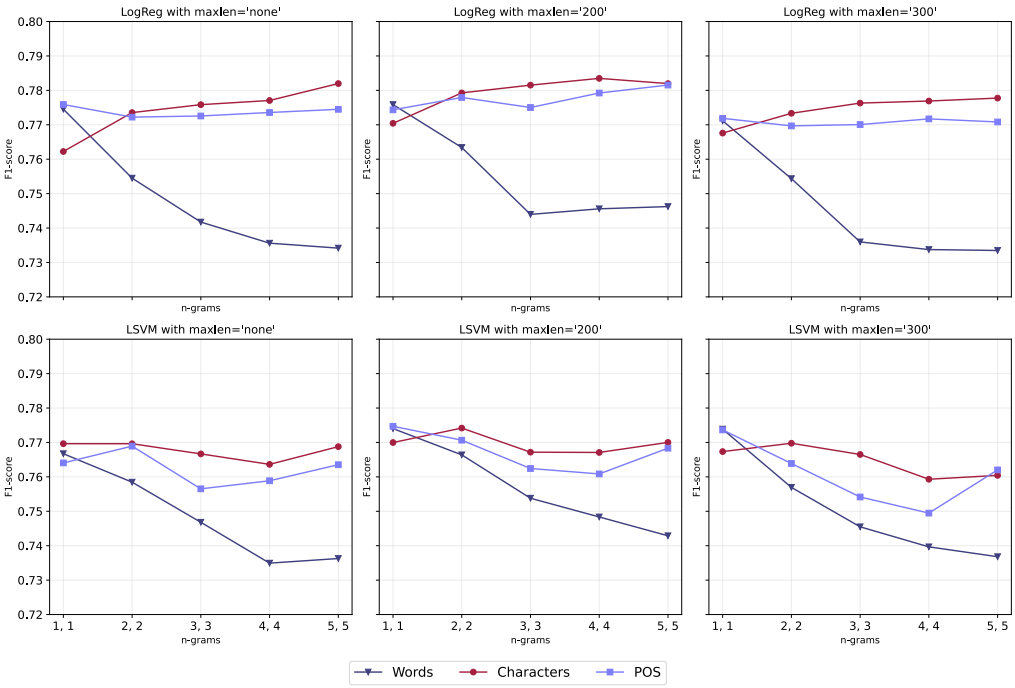
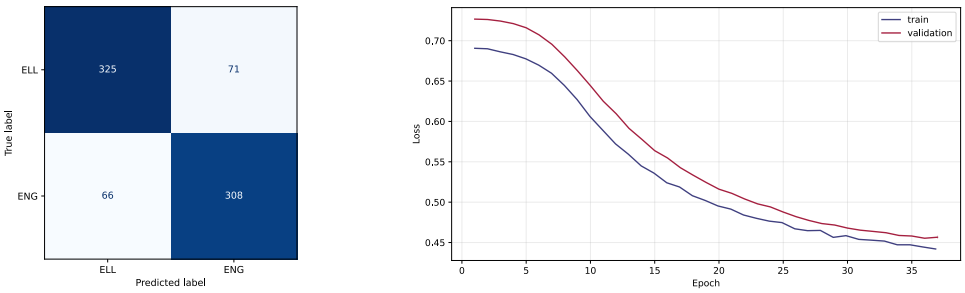


Figure 5: Comparison of the F1-macro scores of the two statistical classifiers, Linear SVM and Logistic Regression, per feature set (words, characters, POS tags), n-gram ranges (individual) and maximum text length (ℓ_{max}).

in the same language but produced with a different L1 in mind. Overall, the strong performance of BERT-based models in text classification tasks suggests that they may be a valuable tool for NLI tasks as well. Finally, these models were also substantially easier to set up and train, since no extra features were fed into them apart from the plain texts.



(a) **(b)**

Figure 6: Confusion matrix and losses of the DistilBERT$_{\text{uncased}}$ model in 37 epochs.

4.2 Ensemble Classifier

Training ensembles is a common practice that aims at improving the final results by combining the best parts of the models that comprise them. However, this comes with the opposite risk, as well. In fact, this is what happened in our case; Two components of the ensemble, the classifiers trained with the linguistic branch of features, were underperforming compared to when they were trained with the same features and grid search hyperparameters individually (73.5% compared to 78.3% for the Linear SVM, and 72.6% compared to 78.6% for the Logistic Regression classifiers).

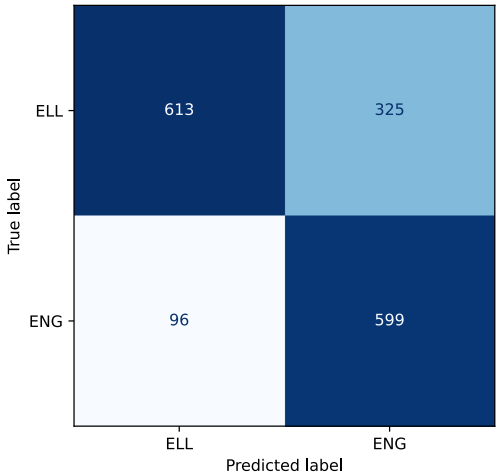


Figure 7: Confusion matrix for the statistical ensemble model with hard voting.

What strikes the most, according to our opinion, and is worth of noting is that the word embeddings predicted correctly 149 samples that the other two, linguistic-features employed estimators predicted wrong and, thus, due to the majority voting pipeline, the wrong label was assigned. If we take a look at the opposite case, namely correct predictions by both linguistic estimators but wrong by the word embeddings LogReg, we are (not) surprised to see that there are only 4 such cases. The predominant position of (recently released) pre-trained word embeddings in (binary) text classification is clear. This provides us with the great opportunity to run quickly through the wrong predictions made by the model and try to identify some of the patterns that may have caused this decreased performance that ultimately classifies the ensemble as an unfruitful attempt.

4.3 Precision and Recall Trade-off

The classification report provided below originates from the ensemble model, which we analysed in detail in Section 3.3.3, and suggests that the classifier has a higher precision for the `ELL` class and a higher recall for the `ENG` class. This is found in all the experiments that we ran, regardless of feature set, sample size (ℓ_{max}), and all reported scores. This suggests that there is a tendency in the classifiers being more confident and accurate in

their predictions for ELL, but at the same time are also more likely to identify instances of ENG even if they are not as confident in those predictions. One potential explanation for this pattern of results could be the characteristics of the two classes. It is possible that the features or patterns that are associated with ELL are more distinct or easier to identify than those associated with ENG, which could lead the classifier to be more accurate in its predictions for ELL. On the other hand, the features or patterns associated with ENG may be more subtle or less distinctive, which could make it more challenging for the classifiers to accurately identify samples deriving from the ENG subset. It may be useful to further analyse the characteristics of the two classes and the behaviour of the classifiers to better understand the reasons that led to these performance differences. We will do so in the next section.

		ELL		**ENG**	
Model	Features set	Precision	Recall	Precision	Recall
LogReg	Linguistic	**0.82**	0.67	0.71	**0.84**
	Embeddings	**0.89**	0.67	0.67	**0.89**
LSVM	Linguistic	**0.79**	0.67	0.70	**0.82**
	Embeddings	**0.92**	0.59	0.63	**0.93**
Ensemble	All	**0.83**	0.65	0.68	**0.82**

Table 5: Precision and recall scores for some exemplary classifiers, each with a different combination of features and algorithm used. Precision is always higher for ELL, and so is the recall but for ENG (both denoted in bold).

4.4 Error Analysis

For the last section of our analysis of the overall results, we gathered all predictions made by the individual statistical classifiers and the ensemble, as a means of diving deeper into what went wrong and the models underperformed compared to earlier experiments. We found 3 interesting patterns we would like to elaborate on and dedicate the final lines of the analysis in this study.

We will focus on the samples that all three models predicted the wrong label for, as these should draw our greatest attention. These samples account for 233 texts in total, which translates to approximately 14.2% of the testing data (in the statistical branch) and correspond to a larger amount than the 2-vs-1 wrong predictions (11.6%), given that the total accuracy score is 74.2%.

First and foremost, we extracted some preliminary information about the wrong-predicted samples, such as length and total number of punctuation marks. Although the latter does not seem to provide us with any useful information, if we visualise the former on a box plot, a pattern arises: 75% of these texts are of length 90 words at the most. The mean of the texts is just below 60 words. We can therefore assume, with certainty to some extent, that the label of short-length texts is more challenging for a classifier to predict correctly. This is not uncommon in NLP tasks, as we have already discussed in detail in Section 2.1 and Section 3.1, and when testing the impact of different values for ℓ_{max} (Section 3.1.1).

Second, in most of the incorrectly-predicted texts, words that are more common in the other class are present. The ones that the models encountered the most in such cases are places and names known in the countries with the different L1s (e.g., *Aegean (sea)*, *British*, *American*) and country-specific topics (e.g., *austerity*, *economy*, *guns*) that were written by authors who belong to the opposite class, however. Let us take a look at some examples:

Text	Pred.	True
*Just to give you people a sense of the utter *unreality* that is being constantly spewed towards the Greek population from the Greek establishment.*	ELL	ENG
If the Greek economy doesn't achieve a long-term growth rate significantly higher than the average of the EU, the country won't be able to catch up with the other EU members	ELL	ENG
It's with particular joy that I inform you, that all members of the Greek delegation who visited your institute on 00 May 0000 and worked with you on the conference on Greek-Russian relations, have taken the following positions in the Greek Government	ELL	ENG
It is obvious that the "greatness" of a film is a subjective matter and that I am probably the only human that has this opinion (even in Greece most people have never even heard about the movie).	ELL	ENG

Table 6: Examples of testing data samples that were misclassified by all models of the ensemble. Words that seem to be more common in L1 Greek texts are present, making it a valid hypothesis as to why the model might have made incorrect predictions.

Third but, in our opinion, the most interesting observation is that in the texts that were written by L1 English authors but were identified with the ELL class words of Greek origin are present. Although the impact of cognates has been previously explored for the NLI task, we cannot be sure that is true for the samples at hand. However, we should clarify that it is very common for L1 speakers of Greek to prefer the Greek-etymology word rather than the equivalent in English that comes from other languages. Rabinovich, Tsvetkov, and Wintner (2018) give the example of *heaven*, that Greek L1 speakers tend to use *paradise* since it comes from the Greek word παράδεισος (/parádeīsos/). We list some of them below (Table 7) and denote the Greek-originated words in bold.

Unfortunately, we cannot be sure if any of the above factors actually contributed to the predictions the wrong subset thereof could be attributed – even partially – to these. However, it makes for an interesting case, and it goes to show that further steps could have been taken to eliminate these biases.

5 Conclusion

The aims of this paper, as enumerated in Section 1.4.1, were (a) to lay the foundations for performing the Native Language Identification (NLI) task for the Greek language for the first time; and (b) to approach it using different machine-learning methods to study which one yield the highest results. Given the aforementioned sections, starting with

Text	Pred.	True
*The whole point to discuss it in terms of "free market" terms is a bit **hypocritical**, except if we are just willfully accepting **propaganda** by whomever each time the article supports.*	ELL	ENG
*I don't blame Europe fully but to act they have not played a negative **role** in our development as a nation or our current **economic** situation is ignorance at best, I mostly blame the Greek people though, **apathy**, etc.*	ELL	ENG
*I can't form an opinion or claim that Albanians have been here for 2 milleniums solely by linguists hell even indo-european langue and **theory** in general is disputed.*	ELL	ENG
*I do believe though that Germany was the first to spread the "lazy Greek" **stereotype** and paint itself as a savior.*	ELL	ENG

Table 7: Examples of testing data samples that were misclassified by all models of the ensemble as ELL. Words that derive from Greek are present in these examples and are denoted in bold.

an overview of the related work done in the field since the first study (Koppel, Schler, and Zigdon 2005a) that formally introduced NLI to the NLP "tasklist", following with the corpora that have been commonly used to tackle this problem, and finally outlining and describing the various pipelines and features that we used, we are in the position to say that it has been successfully carried out.

The NLI landscape could benefit from studies that attempt to introduce *minor languages* to it, as defined by Radó (1987) and Parianou (2009). The first step should always be to run experiments to establish whether the L1 can be distinguished from another, *major* language (such as English). This challenge can be dealt with, in particular since English texts written by L2 authors are easy to find, especially on social media. However, topic bias is a difficult challenge to overcome and a parameter that needs to be taken seriously. Not only is it difficult to gather data that were written on similar occasions (cf. TOEFL examinations) but equally difficult – if not more – is to verify a person's mother tongue, especially on online platforms that primarily rely on anonymity. Thankfully, the L2-Reddit corpus is based on the appropriate metadata to be quite reliable, and this has allowed us to run our experiments and draw meaningful conclusions about the performance of our models on this dataset.

Our methodological approach was introduced in Section 3. The experiments we carried out were of binary nature (Greek vs English as L1) and consisted of various models, with different features, hyperparameters, and classifiers. We started by implementing a simple linear classifier (SVM), followed by a Logistic Regression, both employing Stochastic Gradient Descent for optimisation. The similar results that both of the models yielded, many times regardless of the different parameters during Grid Search Cross-Validation, did not allow us to select one over the other. Soon after, it was made clear that we needed to increase the results, as the linguistic features we incorporated had started reaching their potential. We experimented with OpenAI's recently-released 2nd-gen pre-trained word embeddings, that reached almost state-of-the-art results when combined with linguistic features – as far as non-neural models are concerned. Finally,

we implemented two pipelines exploiting the power of pre-trained large language models with two BERT-based models that also achieved very high results. Among the most interesting findings is that all models achieved high precision and high recall for the Greek and the English L1 classes, respectively, This confirms our initial hypothesis that the differences at hand can be measured and there are even deeper patterns than the features we extracted on our own from the input data.

Last but not least, we combined the 3 best individual models with 2 sets of features (linguistic and word embeddings) in a hard-voting ensemble classifier, but the results were not up to par, primarily due to the linguistic-feature models underperforming. However, an error analysis on the predictions that all 3 models misclassified gives us some indications as to why that might be the case. These questions could be explored further in the future, along with exploiting new approaches for the NLI task for zero- or low-shot classification for low-data languages or even L2 data. Last but not least, the generative AI landscape, especially in text generation, evolves so rapidly that we might see NLI becoming more mainstream and tailored to our multi-lingual interactions further depending thereon. For the future, it would also be interesting to develop multi-class experiments and include Greek to see whether there are similrities or confusions with other L1s.

Finally, another exciting application for NLI in the future could be to extend its application to the translation field in general. In the translation industry, the use of L1 and L2 languages is analogous to target and source languages, since translators primarily translate from an L2 (source) to their mother tongue (L1), although the opposite is not uncommon, especially in cases from minor to major languages (e.g., from Greek into English). Linguistic quality assessment (LQA), being an essential part of the translation workflow, already makes use of qualitative metrics to provide feedback to a translator's work prior to them submitting their work, such as consistent use of terminology and punctuation, and terminology concordance. The next step would be to use the translators' L1 to infer their proficiency in L2 and ultimately the quality of the translation. NLI could be very helpful in providing quantitative feedback since people are unable to spot what is known in the literature as "translationese" (Gellerstam 1986; Baker, Francis, and Tognini-Bonelli 1993), i.e., linguistic patterns ('artifacts') that seem unfamiliar to the native speakers of the so-called target language. For example, when translators with L1 Greek are asked to translate a text into L2 English, they could be able to get an indication of how "Greek" the translated text is and perform further editing.

There has been a lack of previous research investigating the connection between the NLI task and the professional translation process, although it is an area where it applies every day. That would result in professional linguists being able to harness the potential of Natural Language Processing into getting a deeper understanding of their unconscious choices of words and employment of grammatical and syntactical patterns when asked to translate a text whose source language is their mother tongue into a target language of which they are not native speakers. The study at hand could provide insights into the unique challenges and characteristics of the Greek-English language pair. It goes without saying that, while producing a written (and oral, of course) text in a foreign language, people are unconsciously translating from their mother tongue. There is definitely an analogy of the pair source-target language translating to native-foreign language writing. Bringing NLI and the potential "leakage" of L1-specific patterns into L2 translations could help professionals better understand the specific challenges and nuances of translating between these languages, and by extension enable them to produce more accurate and natural target texts, i.e., translations.

References

Argamon, Shlomo, Marin Šarić, and Sterling S Stein. 2003. Style mining of electronic messages for multiple authorship discrimination. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '03*, ACM Press, New York, New York, USA.

Argamon, Shlomo, Casey Whitelaw, Paul Chase, Sobhan Raj Hota, Navendu Garg, and Shlomo Levitan. 2007. Stylistic text classification using functional lexical features. *Journal of the American Society for Information Science and Technology*, 58(6):802–822.

Baker, Mona, Gill Francis, and Elena Tognini-Bonelli, editors. 1993. *Text and Technology: In honour of John Sinclair*. John Benjamins.

Bjerva, Johannes, 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, Association for Computational Linguistics, Copenhagen, Denmark.

Blanchard, Daniel, 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.

Boumparis, Dimitrios. 2022. Identifying Crosswriters' Altering Style in Books for Children and Adults Using Supervised Machine Learning. Unpublished paper for the 'Computational Literary Studies' course at the University of Antwerp, Belgium.

Boumparis, Dimitrios, Hannes den Ouden, and Ine Gevers. 2022. Offensive Language Identification on Tweets Using Machine Learning and Deep Learning Approaches. Unpublished paper for the 'Natural Language Processing' course at the University of Antwerp, Belgium.

Brooke, Julian and Graeme Hirst. 2012. Robust, lexicalized native language identification. In *Proceedings of COLING 2012*, pages 391–408, The COLING 2012 Organizing Committee, Mumbai, India.

Brooke, Julian and Graeme Hirst. 2013. Native language detection with 'cheap' learner corpora. In *Twenty Years of Learner Corpus Research. Looking Back, Moving Ahead: Proceedings of the First Learner Corpus Research Conference (LCR 2011)*, volume 1, page 37, Presses universitaires de Louvain.

Brown, Peter F., Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. 1992. Class-based *n*-gram models of natural language. *Computational Linguistics*, 18(4):467–480.

Chan, Sophia, Maryam Honari Jahromi, Benjamin Benetti, Aazim Lakhani, and Alona Fyshe. 2017. Ensemble methods for native language identification. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 217–223, Association for Computational Linguistics, Copenhagen, Denmark.

Cimino, Andrea 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, Association for Computational Linguistics, Copenhagen, Denmark.

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

Gellerstam, Martin. 1986. Translationese in swedish novels translated from english. In *Translation Studies in Scandinavia*, page 8895, CWK Gleerup.

Goldberg, Yoav and Graeme Hirst. 2017. *Neural Network Methods in Natural Language Processing*. Morgan amp; Claypool Publishers.

Goldin, Gili. 2019. Native language identification with user generated content. Msc thesis, Department of Computer Science, University of Haifa, Haifa, Israel, 08.

Goldin, Gili, 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, Association for Computational Linguistics, Brussels, Belgium.

Granger, Sylviane. 2003. The international corpus of learner english: A new resource for foreign language learning and teaching and second language acquisition research. *TESOL Quarterly*, 37(3):538–546.

Granger, Sylviane, Maïté Dupont, Fanny Meunier, Hubert Naets, and Magali Paquot. 2020. *International Corpus of Learner English. Version 3*. Presses universitaires de Louvain, Louvain-la-neuve.

Granger, Sylviane, Fanny Meunier, Hubert Naets, and Magali Paquot. 2009. *The International Corpus of Learner English. Version 2. Handbook and CD-ROM*. Presses universitaires de Louvain,

Louvain-la-neuve.
He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition.
Ionescu, Radu Tudor, 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, Association for Computational Linguistics, Doha, Qatar.
Jarvis, Scott, Yves Bestgen, and Steve Pepper. 2013. Maximizing classification accuracy in native language identification. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 111–118, Association for Computational Linguistics, Atlanta, Georgia.
Kestemont, Mike. 2014. Function words in authorship attribution. from black magic to theory? In *Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLFL)*, pages 59–66, Association for Computational Linguistics, Gothenburg, Sweden.
Koppel, Moshe and Jonathan Schler. 2003. Exploiting stylistic idiosyncrasies for authorship attribution. In *Proceedings of IJCAI'03 Workshop on Computational Approaches to Style Analysis and Synthesis*.
Koppel, Moshe, Jonathan Schler, and Kfir Zigdon. 2005a. Automatically determining an anonymous author's native language. In *Intelligence and Security Informatics*, pages 209–217, Springer Berlin Heidelberg, Berlin, Heidelberg.
Koppel, Moshe, Jonathan Schler, and Kfir Zigdon. 2005b. Determining an author's native language by mining a text for errors. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, page 624–628, Association for Computing Machinery, New York, NY, USA.
Kulmizev, Artur, Bo Blankers, Johannes Bjerva, Malvina Nissim, Gertjan van Noord, Barbara Plank, and Martijn Wieling. 2017. The power of character n-grams in native language identification. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 382–389, Association for Computational Linguistics, Copenhagen, Denmark.
Lotfi, Ehsan, 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, International Committee on Computational Linguistics, Barcelona, Spain (Online).
Lynum, André. 2013. Native language identification using large scale lexical features. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 266–269, Association for Computational Linguistics, Atlanta, Georgia.
Malmasi, Shervin and Mark Dras. 2017. Multilingual native language identification. *Natural Language Engineering*, 23(2):163–215.
Malmasi, Shervin, 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, Association for Computational Linguistics, Copenhagen, Denmark.
Malmasi, Shervin, Joel Tetreault, and Mark Dras. 2015. Oracle and human baselines for native language identification. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 172–178, Association for Computational Linguistics, Denver, Colorado.
Markov, Ilia, Lingzhen Chen, Carlo Strapparava, and Grigori Sidorov. 2017. CIC-FBK approach to native language identification. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 374–381, Association for Computational Linguistics, Copenhagen, Denmark.
Markov, Ilia, Vivi Nastase, and Carlo Strapparava. 2018. Punctuation as native language interference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3456–3466, Association for Computational Linguistics, Santa Fe, New Mexico, USA.
Markov, Ilia, Vivi Nastase, and Carlo Strapparava. 2022. Exploiting native language interference for native language identification. *Natural Language Engineering*, 28(2):167–197.
Markov, Ilia, Efstathios Stamatatos, and Grigori Sidorov. 2018. Improving cross-topic authorship attribution: The role of pre-processing. In *Computational Linguistics and Intelligent Text Processing*, pages 289–302, Springer International Publishing, Cham.
de Marneffe, Marie-Catherine, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal Stanford dependencies: A cross-

linguistic typology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4585–4592, European Language Resources Association (ELRA), Reykjavik, Iceland.

Mikolov, Tomás, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546.

Newton, B. 1972. *The Generative Interpretation of Dialect: A Study of Modern Greek Phonology*. Cambridge Studies in Linguistics. Cambridge University Press.

Oh, Yoo Rhee, Hyung-Bae Jeon, Hwa Jeon Song, Yun-Kyung Lee, Jeon-Gue Park, and Yun-Keun Lee. 2017. A deep-learning based native-language classification by using a latent semantic analysis for the NLI shared task 2017. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 413–422, Association for Computational Linguistics, Copenhagen, Denmark.

Parianou, Anastasia. 2009. *Translating from Major Into Minor Languages*. Diavlos, Athens.

Pedregosa, Fabian, 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 Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12(null):2825–2830.

Pennington, Jeffrey, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Association for Computational Linguistics, Doha, Qatar.

Petrov, Slav, Dipanjan Das, and Ryan T. McDonald. 2011. A universal part-of-speech tagset. *CoRR*, abs/1104.2086.

Popescu, Marius and Radu Tudor Ionescu. 2013. The story of the characters, the DNA and the native language. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 270–278, Association for Computational Linguistics, Atlanta, Georgia.

Posadas Durán, Juan, Grigori Sidorov, Helena Gomez Adorno, Ildar Batyrshin, Elibeth Mirasol-Mélendez, Gabriela Posadas-Durán, and Liliana Chanona-Hernández. 2017. Algorithm for extraction of subtrees of a sentence dependency parse tree. *Acta Polytechnica Hungarica*, 14:79.

Rabinovich, Ella, Yulia Tsvetkov, and Shuly Wintner. 2018. Native language cognate effects on second language lexical choice. *Transactions of the Association for Computational Linguistics*, 6(0):329–342.

Radford, Alec, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Radó, György. 1987. A typology of lld translation problems. *Babel*, 33(1):6–13.

Sapkota, Upendra, Steven Bethard, Manuel Montes, and Thamar Solorio. 2015. Not all character n-grams are created equal: A study in authorship attribution. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–102, Association for Computational Linguistics, Denver, Colorado.

Smith, Tony C. and Ian H. Witten. 1993. Language inference from function words. Working Paper.

Stamatatos, Efstathios. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556.

Swanson, Benjamin and Eugene Charniak. 2012. Native language detection with tree substitution grammars. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 193–197, Association for Computational Linguistics, Jeju Island, Korea.

Tetreault, Joel, 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, Association for Computational Linguistics, Atlanta, Georgia.

Tetreault, Joel, 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, The COLING 2012 Organizing Committee, Mumbai, India.

Tsur, Oren and Ari Rappoport. 2007. Using classifier features for studying the effect of native language on the choice of written second language words. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pages 9–16, Association for Computa-

tional Linguistics, Prague, Czech Republic.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Wong, Sze-Meng Jojo and Mark Dras. 2009. Contrastive analysis and native language identification. In *Proceedings of the Australasian Language Technology Association Workshop 2009*, pages 53–61, Sydney, Australia.

Wong, Sze-Meng Jojo and Mark Dras. 2011. Exploiting parse structures for native language identification. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1600–1610, Association for Computational Linguistics, Edinburgh, Scotland, UK.

Wong, Sze-Meng Jojo, Mark Dras, and Mark Johnson. 2012. Exploring Adaptor Grammars for native language identification. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 699–709, Association for Computational Linguistics, Jeju Island, Korea.

Wu, Ching-Yi, Po-Hsiang Lai, Yang Liu, and Vincent Ng. 2013. Simple yet powerful native language identification on TOEFL11. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 152–156, Association for Computational Linguistics, Atlanta, Georgia.

Yannakoudakis, Helen, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Association for Computational Linguistics, Portland, Oregon, USA.

Yarra, Chiranjeevi, Achuth Rao M.V., and Prasanta Kumar Ghosh. 2018. Automatic native language identification using novel acoustic and prosodic feature selection strategies. In *2018 15th IEEE India Council International Conference (INDICON)*, pages 1–6.

Zeman, Daniel. 2008. Reusable tagset conversion using tagset drivers.

Zhao, Ying and Justin Zobel. 2007. Searching with style: Authorship attribution in classic literature. In *Proceedings of the Thirtieth Australasian Conference on Computer Science - Volume 62*, ACSC '07, page 59–68, Australian Computer Society, Inc., AUS.

Zipf, George K. 1949. *Human Behaviour and the Principle of Least Effort*. Addison-Wesley.

Appendix:

Statistical Models Comparison Tables

All following tables show the mean ($\bar{\mu}$) and standard deviation (SD, σ) of the F1-macro scores after 10-fold CV on the final corpus per maximum word length (ℓ_{max}), in binary classification (`ELL` vs `ENG` L1), comparing the two statistical models.

		Linear SVM		**LogReg**	
ℓ_{max}	n-grams	mean ($\bar{\mu}$)	SD (σ)	mean ($\bar{\mu}$)	SD (σ)
200	$n = (1, 1)$	0.616607	0.030103	0.613732	0.026880
	$n = (1, 2)$	0.722078	0.052722	0.712393	0.059836
	$n = (1, 3)$	0.772167	0.024091	0.767290	0.030161
	$n = (1, 4)$	0.779050	0.028560	0.777243	0.034443
	$n = (1, 5)$	0.780841	0.031793	0.783270	0.030817
	$n = (1, 6)$	0.786859	0.027712	0.786692	0.031693
	$n = (1, 7)$	0.790132	0.027455	0.788008	0.032413
	$n = (1, 8)$	0.792095	0.027260	0.785370	0.031607
	$n = (1, 9)$	0.792801	0.027382	0.785874	0.030294
	$n = (1, 10)$	**0.795409**	0.025709	0.785753	0.033439
300	$n = (1, 1)$	0.600811	0.028931	0.589993	0.028804
	$n = (1, 2)$	0.718483	0.038349	0.711903	0.033798
	$n = (1, 3)$	0.761777	0.032398	0.747923	0.033855
	$n = (1, 4)$	0.771229	0.033193	0.768297	0.031859
	$n = (1, 5)$	0.772588	0.031787	0.773632	0.033221
	$n = (1, 6)$	0.778980	0.031625	0.774545	0.029152
	$n = (1, 7)$	**0.780847**	0.028894	0.778770	0.031900
	$n = (1, 8)$	0.778007	0.032135	0.777342	0.030489
	$n = (1, 9)$	0.779635	0.031593	0.775188	0.031121
	$n = (1, 10)$	0.780569	0.031497	0.774895	0.032083
none	$n = (1, 1)$	0.610172	0.041129	0.600360	0.050653
	$n = (1, 2)$	0.731930	0.057442	0.715502	0.054000
	$n = (1, 3)$	0.766412	0.046062	0.762290	0.047345
	$n = (1, 4)$	0.774334	0.047490	0.772625	0.042969
	$n = (1, 5)$	0.779970	0.047326	0.777031	0.046384
	$n = (1, 6)$	0.780676	0.045157	0.781785	0.051249
	$n = (1, 7)$	0.779854	0.042399	0.782427	0.051882
	$n = (1, 8)$	0.783592	0.041463	0.778614	0.051855
	$n = (1, 9)$	0.783224	0.042094	0.779433	0.052917
	$n = (1, 10)$	**0.785806**	0.042092	0.779292	0.050488

Table 8: F1-macro scores per ℓ_{max} for different character n-gram ranges (continuous). Highest scores are in bold.

			Linear SVM		**LogReg**	
ℓ_{max}	Feature	n-grams	mean ($\bar{\mu}$)	SD (σ)	mean ($\bar{\mu}$)	SD (σ)
200	Characters	$n = (1, 1)$	0.769978	0.023487	0.770428	0.027331
		$n = (2, 2)$	0.774178	0.025394	0.779267	0.027604
		$n = (3, 3)$	0.767174	0.024994	0.781519	0.025616
		$n = (4, 4)$	0.767094	0.023817	**0.783484**	0.022300
		$n = (5, 5)$	0.770011	0.028266	0.781971	0.024153
	POS	$n = (1, 1)$	0.774693	0.026639	0.774339	0.029004
		$n = (2, 2)$	0.770661	0.029590	0.777937	0.031645
		$n = (3, 3)$	0.762433	0.029255	0.775019	0.027897
		$n = (4, 4)$	0.760854	0.029124	0.779228	0.028308
		$n = (5, 5)$	0.768335	0.027229	**0.781544**	0.029300
	Words	$n = (1, 1)$	0.774031	0.029857	**0.775871**	0.030108
		$n = (2, 2)$	0.766398	0.026805	0.763405	0.028718
		$n = (3, 3)$	0.753791	0.031193	0.743964	0.029368
		$n = (4, 4)$	0.748353	0.031075	0.745585	0.029295
		$n = (5, 5)$	0.742884	0.034769	0.746232	0.033937
300	Characters	$n = (1, 1)$	0.767345	0.032542	0.767587	0.029265
		$n = (2, 2)$	0.769768	0.028641	0.773338	0.030621
		$n = (3, 3)$	0.766507	0.027666	0.776305	0.029716
		$n = (4, 4)$	0.759308	0.029363	0.776904	0.032194
		$n = (5, 5)$	0.760434	0.027685	**0.777756**	0.029639
	POS	$n = (1, 1)$	**0.773648**	0.027373	0.771870	0.032056
		$n = (2, 2)$	0.763887	0.034457	0.769679	0.034955
		$n = (3, 3)$	0.754155	0.033891	0.770064	0.032533
		$n = (4, 4)$	0.749475	0.029220	0.771715	0.027733
		$n = (5, 5)$	0.762025	0.032657	0.770821	0.026698
	Words	$n = (1, 1)$	**0.773865**	0.029172	0.771164	0.029523
		$n = (2, 2)$	0.756949	0.027414	0.754350	0.035560
		$n = (3, 3)$	0.745487	0.034301	0.735971	0.029188
		$n = (4, 4)$	0.739667	0.030950	0.733744	0.035434
		$n = (5, 5)$	0.736801	0.031337	0.733479	0.035260
none	Characters	$n = (1, 1)$	0.769639	0.049332	0.762245	0.041150
		$n = (2, 2)$	0.769628	0.043965	0.773540	0.046331
		$n = (3, 3)$	0.766676	0.046127	0.775846	0.048555
		$n = (4, 4)$	0.763634	0.047004	0.777058	0.052771
		$n = (5, 5)$	0.768779	0.041639	**0.781975**	0.047952
	POS	$n = (1, 1)$	0.764050	0.048942	**0.775893**	0.051784
		$n = (2, 2)$	0.768926	0.041780	0.772223	0.046841
		$n = (3, 3)$	0.756517	0.041598	0.772559	0.049165
		$n = (4, 4)$	0.758866	0.043127	0.773570	0.048252
		$n = (5, 5)$	0.763531	0.041725	0.774494	0.044486
	Words	$n = (1, 1)$	0.766724	0.048531	**0.774526**	0.044187
		$n = (2, 2)$	0.758459	0.045770	0.754493	0.045856
		$n = (3, 3)$	0.746809	0.050415	0.741737	0.046418
		$n = (4, 4)$	0.734935	0.049830	0.735601	0.050482
		$n = (5, 5)$	0.736278	0.053506	0.734166	0.044102

Table 9: F1-macro scores per feature set and n-gram ranges. Highest scores are in bold.

			Linear SVM		**LogReg**	
ℓ_{max}	Feature	n-grams	mean ($\bar{\mu}$)	SD (σ)	mean ($\bar{\mu}$)	SD (σ)
200	Characters	$n = (1, 1)$	0.767702	0.025151	0.776402	0.024830
		$n = (1, 2)$	0.775488	0.023755	0.779492	0.024261
		$n = (1, 3)$	0.775152	0.023321	0.782263	0.024030
		$n = (1, 4)$	0.773861	0.027301	**0.786204**	0.027782
		$n = (1, 5)$	0.774530	0.022233	0.781660	0.024661
	POS	$n = (1, 1)$	0.774035	0.027871	0.774009	0.026165
		$n = (1, 2)$	0.768710	0.030867	0.775591	0.030674
		$n = (1, 3)$	0.768735	0.033411	0.779069	0.027577
		$n = (1, 4)$	0.765455	0.032099	0.779564	0.028390
		$n = (1, 5)$	0.767813	0.026895	**0.780464**	0.032749
	Words	$n = (1, 1)$	0.773023	0.025804	0.778952	0.028880
		$n = (1, 2)$	0.780928	0.027554	0.776027	0.031392
		$n = (1, 3)$	0.781998	0.026896	0.776267	0.030232
		$n = (1, 4)$	**0.783310**	0.028345	0.766888	0.032431
		$n = (1, 5)$	0.778969	0.027188	0.762573	0.029946
300	Characters	$n = (1, 1)$	0.763342	0.030924	0.768338	0.030002
		$n = (1, 2)$	0.766767	0.027822	0.773281	0.034321
		$n = (1, 3)$	0.771384	0.028097	0.772702	0.026769
		$n = (1, 4)$	0.769881	0.028420	**0.774774**	0.027326
		$n = (1, 5)$	0.768674	0.026666	0.774116	0.036223
	POS	$n = (1, 1)$	0.770206	0.032214	0.768813	0.027884
		$n = (1, 2)$	0.760301	0.035370	0.770949	0.030640
		$n = (1, 3)$	0.759343	0.035130	0.770939	0.032147
		$n = (1, 4)$	0.759183	0.031938	0.773704	0.031835
		$n = (1, 5)$	0.761938	0.033669	**0.776107**	0.032608
	Words	$n = (1, 1)$	0.770400	0.028499	0.773076	0.028434
		$n = (1, 2)$	0.772828	0.025055	0.772493	0.032897
		$n = (1, 3)$	0.774548	0.024733	0.765874	0.037868
		$n = (1, 4)$	**0.778504**	0.027260	0.760484	0.033769
		$n = (1, 5)$	0.773741	0.027464	0.762961	0.036875
none	Characters	$n = (1, 1)$	0.770808	0.050333	0.769604	0.062220
		$n = (1, 2)$	0.772521	0.049833	0.774725	0.052501
		$n = (1, 3)$	0.768064	0.045349	0.779691	0.049781
		$n = (1, 4)$	0.769167	0.046628	**0.780277**	0.050777
		$n = (1, 5)$	0.765865	0.046301	0.777984	0.051720
	POS	$n = (1, 1)$	0.768421	0.047690	0.773917	0.048415
		$n = (1, 2)$	0.763963	0.043064	0.776445	0.052747
		$n = (1, 3)$	0.762714	0.043811	0.774246	0.052301
		$n = (1, 4)$	0.761585	0.040196	0.774300	0.050394
		$n = (1, 5)$	0.764563	0.045198	**0.777925**	0.051743
	Words	$n = (1, 1)$	0.765265	0.048926	**0.777475**	0.049262
		$n = (1, 2)$	0.776876	0.047088	0.772450	0.052114
		$n = (1, 3)$	0.777085	0.047998	0.767211	0.048499
		$n = (1, 4)$	0.776066	0.047676	0.759074	0.048723
		$n = (1, 5)$	0.777287	0.048156	0.755464	0.044807

Table 10: F1-macro scores per feature set and n-gram ranges (continuous). Highest scores are in bold.