Manchester Metropolitan University

e-space

Manchester Metropolitan University's Research Repository

Please cite the Published Version

Sarwar, Raheem, Teh, Pin Shen, Fayyaz, Muhammad Asad Bilal, Sabah, Fahad, Hassan, Muhammad Umair and Hassan, Syed Mustafa (2025) Enhancing Educational Equity: A Native Language Identification Approach for Tailoring Linguistic Support and Inclusive Curricula. In: Research and Innovation Forum 2024, 10 April - 12 April 2024, Ravello, Italy.

DOI: https://doi.org/10.1007/978-3-031-78623-5_19

Publisher: Springer

Version: Accepted Version

Downloaded from: https://e-space.mmu.ac.uk/641879/

Usage rights: © In Copyright

Additional Information: This is an author accepted manuscript of a conference paper published in RIIFORUM 2024 (Springer Proceedings in Complexity series) , by Springer.

Enquiries:
If you have questions about this document, contact openresearch@mmu.ac.uk. Please include the URL of the record in e-space. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from https://www.mmu.ac.uk/library/using-the-library/policies-and-guidelines)

Enhancing Educational Equity: A Native Language Identification Approach for Tailoring Linguistic Support and Inclusive Curricula

Raheem Sarwar[1], Pin Shen Teh[1], Muhammad Asad Bilal Fayyaz[1], Fahad Sabah[2], Muhammad Umair Hassan[3], and Syed Mustafa Hassan[4]

[1] School of Business and Law, Manchester Metropolitan University, United Kingdom, R.Sarwar@mmu.ac.uk,
[2] Faculty of Information Technology, Beijing University of Technology, Beijing, China
[3] Department of ICT and Natural Sciences, Norwegian University of Science and Technology - NTNU, Ålesund, Norway
[4] Computational Aeronautics Lab, School of Interdisciplinary Engineering and Sciences, National University of Sciences and Technology, H-12, Islamabad, Pakistan

Abstract. This paper introduces a solution for native language identification (NLI) on texts written in English, French, and German, offering diverse applications in education. NLI research provides insights into students' linguistic backgrounds, enabling educational institutions to customize materials, assignments, and assessments for individual needs. By identifying students who may benefit from language support, institutions can develop targeted language-specific curricula. Understanding students' native languages also helps educators incorporate relevant cultural references and create a more inclusive learning experience. Furthermore, NLI can guide the creation of targeted training for educators, equipping them with strategies to address language-specific challenges and foster effective communication in diverse classrooms. The proposed NLI approach analyzes text samples in non-native languages, providing a robust solution that captures language usage and production patterns across documents and languages. The approach is supported by three new corpora in German, French, and English and has shown superior performance compared to existing state-of-the-art NLI methods and pre-trained language models like DistilBERT, mBERT, multilingual DeBERTa, and XLM-RoBERTa. This enhanced NLI model contributes to improved cross-cultural communication within academic communities, fostering a more inclusive and supportive environment for students and faculty alike.

Keywords: Enhancing Educational Equity, Second Language, Native Language, Part-of-Speech Tagging, Machine Learning

1 Introduction

The native language identification (NLI) task determines an author's *native language* (L1) based on text samples in a *non-native* language (L2) [1–3]. The L2 learners of different L1 speakers make different errors (spelling and grammatical mistakes) [1]. Understanding these errors and associating them with learners from different L1 backgrounds is essential to provide targeted advice to correct them [1].

This task has diverse and impactful applications in the Education sector. NLI research can provide valuable insights into the language backgrounds and linguistic needs of diverse student populations. This information can be used to tailor educational materials, assignments, and assessments to better suit the linguistic needs of each student. Educational institutions can use NLI to identify students who may benefit from language support programs. NLI can inform the development of language-specific curricula. Understanding the predominant native languages of students can guide the inclusion of relevant cultural references, examples, and materials, fostering a more inclusive and culturally sensitive educational experience. Institutions can use NLI to identify the native languages of their students and provide targeted training for educators. Teachers can develop strategies to address language-specific challenges, implement effective communication methods, and create a supportive learning environment. The NLI can inform educational research and contribute to the development of effective teaching methodologies for linguistically diverse classrooms. NLI can contribute to improved cross-cultural communication within the university community. Awareness of the native languages of students and faculty can facilitate better understanding and collaboration, creating a more inclusive and supportive academic environment. Non-native English speakers outnumber native English speakers at present. Nowadays, people are proficient in more than one language [4–8]

and as a result of work-related immigration, there is a substantial community of people who have learned non-English second languages [3, 9]. It has also been estimated that around 45% of the World Wide Web content is written in non-English languages [9]. Therefore, it is important to measure the applicability of NLI solutions to other languages (Malmasi and Draz also applied investigated the effect of different POS tagsets on English, Chinese, and Italian. Their research, however, is limited to learner corpora).

One of our main objectives is to formulate an NLI solution that can outperform the existing state-of-the-art NLI method [3] in terms of accuracy. To achieve this main objective, we represent each document in the training corpus as a *collection of point sets*. We illustrate this concept using Figure 1. We partition each document in the training corpus into fixed size *fragments*. Each resulting fragment is further partitioned into fixed-size *chunks* (a chunk is a collection of Tokens, where each Token represents the content of a text sample separated by a white space character). Following the document partitioning process, we extract universal *part-of-speech (POS) n-gram* based features from each *chunk*. Specifically, we extract *POS bi-grams* and *POS tri-grams* features from each chunk. As a result, each chunk is represented by a *point*, each fragment by a *point set*, and each document by a *collection of point sets* in a multidimensional space (see Section 4.1 for more details). Following the stated document partitioning process, we can devise the native language identification (NLI) problem as a subsequent *set similarity search* (SSS) problem. Given a query document, we perform document partitioning and feature extraction processes on it. As a result, the query document is transformed into a *collection of query fragments* (i.e., collection of point sets). The main motivation behind the *collection of point sets* document representation model is two-fold:

1. Instead of relying on a single native language prediction, it allows us to generate multiple predictions for a query document (one prediction per query fragment), which improves NLI task accuracy. This is because producing numerous predictions for a query document allows us to eliminate the most speculative assumptions while combining the remainder to get a final forecast encompassing the entire query document (see Section 4.4 for more details).
2. It enables us to apply a variety of set distance measures. Specifically, this collection of point set document representation models enables us to use set distance measures such as the partial Hausdorff distance (PHD) [10] and the modified Hausdorff distance (MHD) [11] which help us capture variations in language structures used within and across the documents (see Section 4.2 for details).

For each query fragment, we identify *stylistically similar fragments (SSFs)* from the training corpus. Specifically, we identify top-k SSFs for each query fragment using a set distance function (e.g., MHD or PHD) as a proximity measure between two point sets. In addition, for each fragment of the query document, we run an independent *set similarity query*. The result of a set similarity query is a *set of top-k SSFs*. To generate a *probabilistic prediction* for each query fragment of a query document we apply the PkNN classifier [12] on the retrieved set of top-k SSFs (the motivation for using PkNN is given in Section 4.3). Finally, we aggregate a percentage (i.e., 50%) of most certain query fragment predictions to make a final prediction that represents the entire query document (Section 4 Section 4 contains a detailed description of each step of our solution).

Another main research objective of this investigation is to formulate an NLI solution that can be applied to different languages. We use universal *Part-of-speech* (POS) n-grams features to conduct such a multilingual study for the following reasons. The POS tagging is the core component of most NLI systems (see Section 2 for details). We note that the POS in monolingual NLI research can be estimated using the best POS tagger available for the language in question. For example, in the English language, *Penn Treebank* classifies words into 36 linguistic categories [3]. Similarly, for the French and German languages, *French Treebank* and *Stuttgart/Tübinger* can be used. These two POS taggers classify words into 30 and 55 linguistic categories, respectively [3, 13]. On the other hand, while designing an NLI solution that applies to multiple languages, we need to consider that different granularity of the *linguistics categories* for different languages implies that they are not directly comparable, i.e., *English*, *French* and *German* have 36, 30, and 55 *linguistic categories* , respectively. These various linguistic categories of each language can be converted into a common set of linguistic categories shared by all languages, making experimental results comparable across distinct languages. To perform such multilingual research, we use those POS categories which are common among different languages (see Section 4.1 for details).

We have performed extensive experimental studies to evaluate our solution. Specifically, we compare our solution (SS-NLI) against the existing state-of-the-art NLI method (Comp-NLI) [3]. We also formulate an improved variant of the existing state-of-the-art NLI method called Comp-NLI-E

and compare its performance against our solution. In addition to this, we have compared the performance of our solution against machine learning methods that have been extensively employed in recently published studies (detailed descriptions of each method are given in Section 5.2).

Our main contributions to this investigation can be summarized as follows.

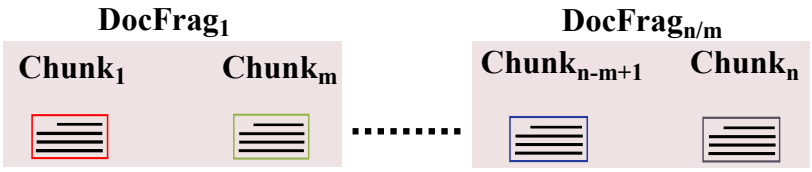


Fig. 1. Document representation model.

- To improve NLI's accuracy, we represent each document as a collection of point sets. This document representation model allows us to remove the most uncertain predictions from a query document and combine the rest of the predictions to produce a single prediction for the entire query document. Furthermore, our solution can recognize variations in language structures used within and across documents.
- We investigate the effectiveness of our document representation model to capture the overlapping information between feature spaces. Specifically, we combine universal POS bi-grams with universal POS tri-grams features set into one feature vector and compare its performance against individual feature sets (i.e., POS bi-grams and POS tri-grams). We found that the combined feature set outperforms the individual feature sets.
- We have performed extensive experimental studies to compare our solution against the competitive techniques. We specifically compare the performance of our solution to that of an existing state-of-the-art NLI method [3]. In addition to this, we have compared the performance of our solution against machine learning methods that have been extensively employed in recently published state-of-the-art studies.
- We also formulate an improved variant of the existing state-of-the-art NLI method called Comp-NLI-E and compare its performance against our solution.
- Based on the document representation model and universal POS categories, we propose a native language identification (SS-NLI) solution, and we demonstrate that our solution can be applied to multiple languages and that it outperforms existing state-of-the-art NLI approaches and pre-trained language models such as DistilBERT [14], mBERT [15], multilingual DeBERTa [16], and XLM-RoBERTa [17].

As for the rest of the paper, Section 2 reviews previous NLI studies. The problem formulation and solution overview is presented in Section 3. Section 4 details our solution. We report the findings of our experimental studies in Section 5. Section 6 contains the conclusions of this investigation and sets the future work directions.

2 Literature Review

Artificial intelligence (AI) and machine learning (ML) have revolutionized various fields, including healthcare, finance, transportation, retail, entertainment, and more, with applications such as medical imaging analysis, autonomous vehicles, and personalized recommendations. Additionally, AI aids customer service, manufacturing, agriculture, security, and language translation, offering efficient and advanced solutions across different industries [18–33]. In this section, we review existing AI and ML techniques used to perform the author profiling and native language identification tasks. Author profiling aims to identify an author's demographic features by analyzing his text samples [34–37]. *Profiling authors* based on their *native languages* (L1) has received extensive attention in recent years [3, 34, 38–42]. The NLI task requires documents written in L2 of the authors labeled with their L1s. The NLI works by identifying the common *language usage patterns* for each group of authors sharing the same native language. Existing NLI studies have used several types of features, which can be categorized broadly as follows.

- **N-grams Based Features.** An n-gram can be defined as a contiguous sequence of n items, such as *characters, words*, or *part-of-speech* (POS) from a text sample. The POS are *linguistic*

categories of words in a text that indicate their syntactic functions. The basic linguistic categories associated with words are *nouns, verbs* and *adjectives*. These types can be further expanded to include morpho-syntactic information. POS n-grams identify the *local syntactic patterns* of language usage in documents written in an L2, which can be used to distinguish groups of authors based on their native languages [2, 43, 44].
- **Function Word Features.** A function word expresses the structural or grammatical relationships between words in a sentence, for example, conjunctions (e.g., but, and), determiners (e.g., the, that), pronouns (e.g., they, she), and prepositions (e.g., in, of) [2, 45–47].
- **Idiosyncratic Features.** Idiosyncratic features are related to language usage anomalies such as grammatical mistakes and misspellings [2, 4].

These aforementioned features have been extensively investigated in several existing NLI studies [2, 3, 43, 45, 47, 48]. For example, [43] performed the pioneer NLI study using function words, and *part-of-speech* (POS) features on a corpus containing samples from 31 Japanese and 6 Chinese native authors. Later on, [2] perform NLI on text samples from five groups of non-native English authors (Czech, Bulgarian, French, Spanish and Russian) and reported 80 % accuracy. They employed several types of features including character n-grams, POS bi-grams, function words, and idiosyncratic features.

As for the classification methods, the most commonly used classifiers in recently published state-of-the-art NLI studies are *support vector machines* (SVM), followed by *maximum entropy analysis* (MaxEnt) and *linear discriminant analysis* (LDA) [45, 47–51]. In this investigation, we compare the performance of our solution against the performance of those classifiers that have been extensively used in recently published state-of-the-art NLI studies (see Section 5.2 for a detailed description of these classifiers).

2.1 Part of Speech tagging and Native Language Identification

In this investigation, we aim at formulating a *native language identification* (SS-NLI) solution that can outperform existing state-of-the-art NLI solutions. The different granularity of linguistic categories for different languages implies that these languages are not directly comparable (i.e., English, French and German have 36, 30, and 55 linguistic categories, respectively) [3]. However, to make different languages comparable using POS-based features, one can introduce linguistic categories that are common among different languages (see Section 4.1 for more details).

Malmasi and Draz proposed a *native language identification* method [3] and measured its performance on multiple languages. This method is based on linear *support vector machines (SVM)* classifier and they used several types of features including POS uni-grams, POS bi-grams, POS tri-grams, and function words. They have shown that the POS tri-grams outperform the other types of features for the NLI task. However, the accuracy level of this existing state-of-the-art NLI method can still greatly be improved. For instance, the existing state-of-the-art NLI method has reported an accuracy level of less than 60% using universal POS tags [3]. Consequently, in this investigation, we propose an NLI solution that improves the accuracy of the NLI task.

Comparison with our solution. Our solution (SS-NLI) outperforms the existing state-of-the-art NLI method (Comp-NLI) [3], because, unlike Comp-NLI, our solution is capable of (i) producing multiple predictions for a query document which enable us to remove uncertain predictions of a query document before combining them to produce a single prediction for the entire query document, and (ii) capturing the variations in language structures used within and across documents. As a result, our solution outperforms the existing state-of-the-art NLI method.

2.2 Summary

Most NLI studies have reported that POS n-grams outperform other types of features, so they are considered a core set of features to perform the NLI task. Most commonly used types of *classification methods* in recently published NLI studies are the *support vector machines* (SVM), followed by *maximum entropy analysis* (MaxEnt) and *linear discriminant analysis* (LDA) [45, 47–61]. However, most of these NLI studies have shown that SVM outperforms other classifiers [3, 62]. Though several researchers have extensively investigated the NLI task, most of the existing NLI studies reported have focused on English corpora. However, nowadays, people are trying to be proficient in more than one language [4, 5, 63–65]. For example, according to [9] , more than half of the world's population is fluent in more than one language. Similarly, the European Union has reported that, on average, 94.5% secondary school students learn more than one language [66]. Moreover, it has been reported that around 45% of web content is written in non-English languages [9]. Thus, there is a substantial need to formulate an effective NLI solution to improve the performance of NLI.

3 Problem Formulation and Solution Overview

In this section, we redefine the *native language identification* (NLI) into probabilistic NLI. For ease of exposition, we first provide the overview of our solution using figure 2. The proposed solution has four stages. (i) preprocessing; (ii) set similarity search; (iii) probabilistic k nearest neighbor (PkNN) classification; and (iv) prediction aggregation.

In the preprocessing step, we perform the document partitioning and feature extraction processes. Specifically, we perform partitioning of each document into fixed-size *fragments*. Each resulting fragment is further partitioned into fixed-size *chunks*. Following document partitioning, we extract two types of attributes from each chunk. These features are based on *part-of-speech (POS)* n-grams. Specifically, we extract (i) *POS bi-grams*; and (ii) *POS tri-grams* from each chunk. As a result, in a multidimensional space, each *chunk* is represented by a point and each *fragment* by a *set of points*. Following the stated document partitioning process, we can devise the native language identification (NLI) problem as a subsequent *set similarity search* problem.

1. We apply the preprocessing step of our solution to a given query document. That is, we partition a query document into fixed-size fragments. Each resulting query fragment is further partitioned into fixed-size chunks. Following the query document partitioning process, we extract two types of features (i) POS bi-grams; and (ii) POS tri-grams from each chunk. Consequently, each chunk is represented as a point and a fragment is represented as a set of points in a multidimensional space.
2. For each query fragment of a query document, we identify *stylistically similar fragments (SSFs)* from the corpus. Specifically, we identify top-k SSFs for each query fragment using *modified Hausdorff distance* [11] as a proximity measure between two point sets. As stated before, we execute an individual *set similarity query* for each fragment of the query document. For instance, we conduct m independent set similarity queries if a query document provides m query fragments after the document segmentation and feature extraction processes.
3. The result of a set similarity query is a set of top-k SSFs. We then apply the PkNN classifier [12] to the retrieved set of top-k SSFs to produce a *probabilistic prediction* for each query fragment of a query document.
4. Finally, we aggregate all query fragment predictions to make a final prediction that represents the entire query document. To clearly describe the process of *fragment prediction aggregation*, we redefine the native language identification problem into a *probabilistic native language identification* problem. The main idea is that for each query fragment of the query document, instead of making one single native language prediction, we produce a probabilistic prediction over a set of native languages as follows.
   Probabilistic NLI. Probabilistic NLI attempts to identify the *native language likelihood* by calculating the PMF (Probability Mass Function) over a *set of likely native languages* of a query fragment.

The following subsections provide a detailed discussion of each step of the proposed solution.

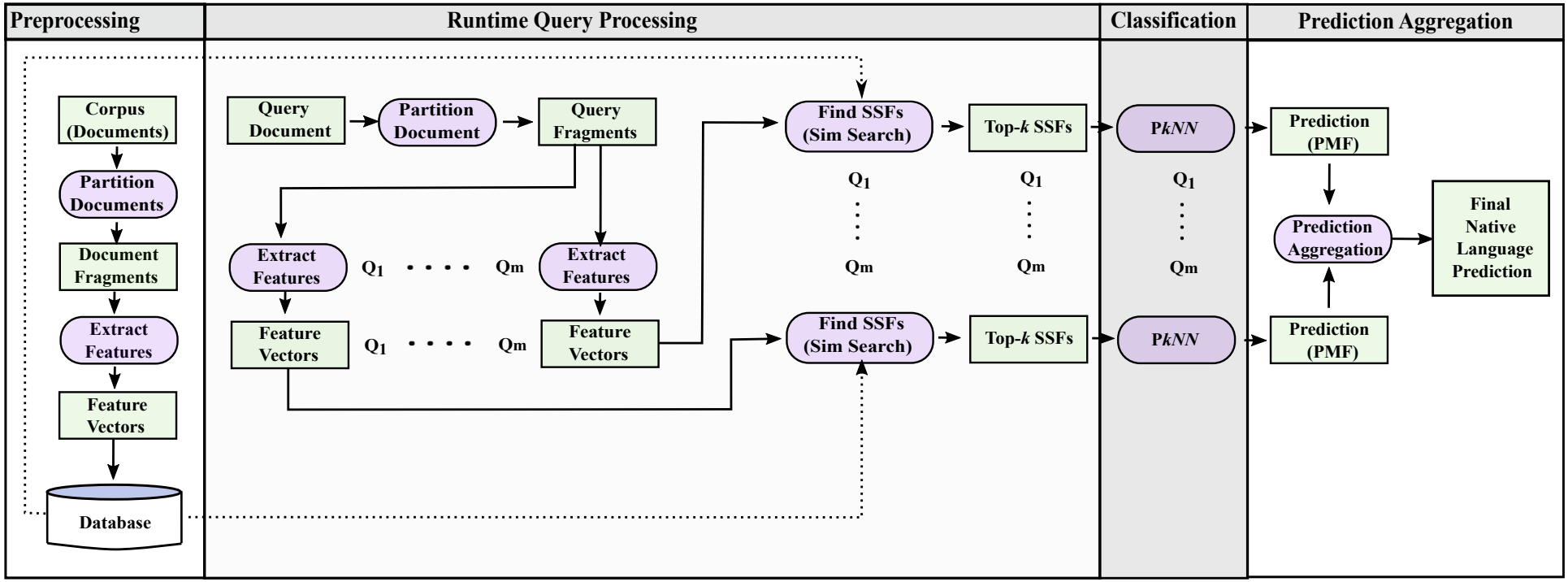


Fig. 2. System overview.

4 Proposed Solution

In this section, we discuss each of our four-step solutions in detail.

4.1 Preprocessing

In our solution, the preprocessing step is in charge of performing *document partitioning* and *feature extraction* processes. In the document partitioning process, we partition each document into fixed-size *fragments*. Each resulting fragment is further decomposed into fixed-size *chunks*. Following the partitioning of the document, we extract *part-of-speech (POS) n-gram* based features from each *chunk*. As a result, each chunk is represented as a *point*, each fragment as a *point set*, and each document as a *collection of point sets*. Recall that the main motivation behind *collection of point sets* document representation model is two-fold:

1. Instead of relying solely on one native language prediction, it allows us to generate numerous predictions for each query document, which improves accuracy. This is because making multiple predictions for a query document allows us to remove a percentage of uncertain predictions before combining them to make a final prediction that represents the entire query document,
2. It enables us to apply a variety of set similarity functions. This *collection of point set* document representation models, in particular, allows us to apply set similarity functions with outlier handling mechanisms, such as *partial Hausdorff distance* [10] and *modified Hausdorff distance* [11], to improve the NLI task's performance.

We set the sizes of a fragment and a chunk to 15000 and 1500 tokens (A text sample's content separated by a white space character), respectively, to obtain reliable stylometric information. Following the partitioning of the document, we extract two types of attributes from each chunk. These attributes are based on *part-of-speech (POS)* n-grams. Specifically, we extract (i) *POS bi-grams*; and (ii) *POS tri-grams* from each chunk.

Algorithm 1 shows the preprocessing step of our solution. The algorithm iterates through all the documents in the corpus. Specifically, the procedure $Partition(d, L)$ partitions each document d of the corpus into fixed size $Fragments$, where L denotes the size of each resulting fragment F (Line 4). The Algorithm 1 next iterates through all resulting $Fragments$ of a document d and further partitions each fragment F into fixed-size chunks, where l denotes the size of each resulting chunk C (Lines 5 to 6). Next, the Algorithm 1 iterates through all resulting $Chunks$ of each fragment F and extracts features from each chunk C (Lines 7 to 8). Finally, each chunk is represented as a point in multidimensional space, whereas each fragment is represented as a cluster of points.

Part-of-Speech Features. *Part-of-speech* (POS) tagging is the core component of most NLI systems as illustrated earlier in Section 2. POS refers to *linguistic categories* in a corpus according to a specific part of speech, based on the definition of the word and its context, e.g., nouns, verbs, and adjectives. Several *tagsets* have been developed for many languages over the past few years. Each of these tagsets is distinct and designed for a specific language. We note that these tagsets can also differ in their levels of granularity within the same language and across the languages. That is, the size of these tagsets varies according to their amount of syntactic classification while giving varying quantities of syntactically significant information [3]. These tagsets are classified into two categories. In discriminating between syntactic categories, they can be fine-grained or coarse-grained. More *morpho-syntactic* information, such as number, gender, person, and verb transitivity, is contained in fine-grained tagsets. A coarser-grained tagset, on the other hand, contains wider syntactic categories such as noun and verb [3].

As reported in many previous NLI studies, the n-grams extracted from these POS tags can discriminate between patterns of language usage among different groups of authors with different native languages. In most of the NLI studies, the POS n-grams are considered to be a core set of features to perform the *native language identification* task (see Section 2).

We will now discuss our feature extraction process in detail. When conducting a monolingual NLI study, the best POS tagger available for a given language can be used to calculate POS. For example, for the English language, one can use *Penn Treebank* which classifies the words into 36 linguistic categories [3]. Similarly, for the French and German languages, one can use *French Treebank* and *Stuttgart/Tübinger* which classify words into 30 and 55 linguistic categories respectively [3, 13, 67]. However, when designing an NLI solution, we must keep in mind that the different granularity of the *linguistics categories* for different languages implies that they are not directly comparable, i.e., 36, 30 and 55 *linguistic categories* for *English*, *French* and *German* languages, respectively. These various linguistic categories of each language can be converted into a common set of linguistic

categories shared by all languages, making experimental results comparable across languages. We used *Universal POS tagger* [68] to conduct such a multilingual study. The *Universal POS tagger* categorizes words into 12 linguistic categories (Verb (verbs), Noun (nouns), Adv (adverbs), Adj (adjectives), Det (articles and determiners), Pron (pronouns), Num (numerals), Adp (prepositions and postpositions), Prt (particles), "." (punctuation marks), Conj (conjunctions) and X (all other categories such as foreign words or abbreviations.)) which are universal across different languages in our corpora. Based on the identified *linguistic categories*, we calculate POS bi-grams and POS tri-grams from each chunk as shown in Algorithm 1.

Algorithm 1 Preprocessing

```
1: procedure PREPROCESSING(d)
2:     L ← 15,000 tokens
3:     l ← 1,500 tokens
4:     DocumentFragments ← Partition(d, L)
5:     for F in DocumentFragments do
6:         FragmentChunks ← Partition(F, l)
7:         for C in FragmentChunks do
8:             Vectors[i] ← CalculateFeatures(C)
9:         end for
10:    end for
11:    return Vectors
12: end procedure
```

4.2 Set similarity Search

Algorithm 2 shows the *set similarity search* step of our solution. While processing a given query document $\mathcal{Q}$, we first apply our solution's preprocessing step (Line 2) to $\mathcal{Q}$. Recall that, the preprocessing step of our solution contains two processes (i) document partitioning; and (ii) feature extraction. Specifically, we partition $\mathcal{Q}$ into fixed-size query fragments. We then partition each query fragment Q into fixed-size chunks. Once the query document partitioning process is completed, we extract two types of features from each chunk as explained in Section 4.1. As a result, the $\mathcal{Q}$ can be represented as a set of point sets (collection of fragments) in a multidimensional space where each point set (fragment) contains a fixed number of points. We conduct an independent set similarity query for each Q in $\mathcal{Q}$ to extract the top-k SSFs from the corpus after we complete preprocessing. It is worth noting that we run a separate *set similarity query* for each fragment of the query document. For instance, we conduct m independent set similarity queries if a query document provides m query fragments after the document segmentation and feature extraction processes (Lines 3 to 5).

We experimented three set similarity techniques while retrieving the top-k SSFs:(i) *standard Hausdorff Distanct (SHD)*, (ii) *partial Hausdorff Distanct (PHD)* [10] and *standard Hausdorff Distanct (MHD)* [69] as a proximity measure between two point sets.

Assume there are two point sets; Q and F. The SHD between these two points sets can be calculated by following steps;

1. arranges all data points in a query fragment Q in order of their shortest distance to the fragment F, and
2. selects the utmost of the shortest distances.

According to many previous studies, SHD is sensitive to outliers. To address the outlier sensitivity issue associated with SHD, researchers developed two SHD versions:*modified Hausdorff distance (MHD)* [69] and *partial Hausdorff distance (PHD)* [10]. Dubuisson et al. [69] proposed MHD, which averages out the effect of the outlier over the minimum distances. The MHD calculation process is given in Algorithm 3. Assume Q and F are two-point sets. The MHD can be calculated by following these steps.

1. rank all data points in a query fragment Q based on the shortest distance to the fragment F; and
2. calculate the average of the shortest distances within a given range i.e., $(50\%, 100\%]$ [69, 70] (cf. Algorithm 3).

Huttenlocher et al. [10] proposed PHD, which considers the top $K\%$ distance values as outliers and discards them from the distance calculations. Section 5.3 reports on the experimental results for set distance measures.

Algorithm 2 Set Similarity Search

```
1: procedure SetSimilaritySearch(Q)
2:     QueryFragments ← Preprocessing(Q)
3:     for Q in QueryFragments do
4:         SSFs.top-k ← MHD(Q, DocumentFragments)
5:         top-k SSFs ← SSFs.Top-k
6:     end for
7:     return top-k SSFs
8: end procedure
```

Algorithm 3 MHD calculations

```
1: procedure MHD(Q , F)
2:     MinDists ← []
3:     MHDDist ← 0
4:     for q in Q do
5:         dmin ← ∞
6:         for f in F do
7:             dist ← dist(q, f)
8:             if dist < dmin then
9:                 dmin ← dist
10:            end if
11:        end for
12:        MinDists.Append(dmin)
13:    end for
14:    MinDists ← Sort(MinDists)
15:    MHDDist ← ComputeAvg[MinDists, (50%, 100%]]
16:    return MHDDist
17: end procedure
```

4.3 Probabilistic-k Nearest Neighbor Classification

To generate a probabilistic prediction for each query fragment, we use a probabilistic k-nearest neighbor classifier (PkNN) with a radial basis function (Gaussian) kernel [12] on the retrieved top-k SSFs in this step.

A simple method to make a probabilistic prediction using PkNN is to count the number of objects (Fragments) normalized with k value. As a result, we get a prediction which is *probability mass function* (PMF) over the set of all the classes of kNN set (i.e., set of top-k SSFs in our case). The PMF measured by this method can be formally expressed as follows:

$$p(y|x, D) = \frac{1}{K} \Sigma_{j \in neighbor(x,K,D)} I(y = y_j). \quad (1)$$

There are two main problems associated with the aforementioned PMF calculation method:

- The first problem is that the probability values associated with each class are proportional to their object *frequencies*. Consequently, less frequent classes result with negligible probability [12]. In order to handle this issue, an exponential function (γ) can be applied to soften the probability distribution. Setting a high value of the exponential function (γ) (e.g., close to 100) produces *spiky* probability distribution over classes. Contrarily, decreasing the γ value uniforms the probability distribution.
- The second problem is that the distances of all candidate objects (Fragments) are ignored. As a result, a large k is required to obtain reliable statistics. To solve this problem, a weight function (ω) can be applied based on the distance of the objects (Fragments) from the query fragment. Candidate fragments with a large distance from query fragments will have less weight.

By incorporating the wight function (ω) and exponential function (γ) into the aforementioned frequency-based method results with the following expression.

$$p(y|x, D, K, \gamma) = \frac{exp[(\gamma/K)\Sigma_{j\sim x}\omega(x, x_j)I(y = y_j)]}{\Sigma_{y'}exp[(\gamma/K)\Sigma_{j\sim x}\omega(x, x_j)I(y' = y_j)]} \tag{2}$$

In order to obtain the desired performance, we conducted several experiments to identify the values for ω and γ. PkNN was chosen because it requires little or no training and achieves classification by comparing instances stored in memory rather than using a generalized model [70]. As a result, no information is lost due to generalization, and new data can be incorporated at any time during the process [70]. Moreover, this classification technique can employ a complex target function [70]. Besides that, due to the non-parametric nature of this classifier, prior knowledge regarding the probability distribution is not required to perform classification. In addition, PkNN allows us to apply a wide range of set similarity functions, including ones with outlier control procedures like MHD, to increase the NLI task's performance.

4.4 Prediction Aggregation

Prediction aggregation is the last step of our solution. As previously stated, each fragment of the query document is executed to an independent *set similarity (SS)* query. For instance, after performing the *preprocessing* step of our solution on the query document $\mathcal{Q}$, if $\mathcal{Q}$ is partitioned into 4 query fragments Q, we execute 4 independent *SS* queries. Consequently, we make one *probabilistic prediction* for each query fragment Q with the help of PkNN classifier. The final step of our solution is to combine all of the *probabilistic predictions* into a single native language prediction for the entire $\mathcal{Q}$. To accomplish this, we simply take the average of all probabilistic predictions. However, not all probabilistic predictions of a query document $\mathcal{Q}$ (one for each query fragment) are equally valuable; for example, some predictions may be extremely uncertain. Incorporating extremely uncertain predictions into the final $\mathcal{Q}$ prediction can harm the overall result. At this point, we use *entropy* as an uncertainty metric to detect and exclude uncertain predictions from the prediction aggregation process. Specifically, we make use of the most certain κ% prediction only in the process of prediction aggregation. The average PMF of the most certain κ% prediction is used to compute the final probabilistic prediction of the entire $\mathcal{Q}$. Table 1 shows an example of this process.

Assume a query document $\mathcal{Q}$ yielded four query fragments Q_1, Q_2, Q_3 and Q_4. We run four separate set similarity queries, one for each fragment (cf. the 1_{st} column in Table 1). As a result, for each query fragment, we get one probabilistic prediction (e.g., Q_1: $[G : 0.33,\ H : 0.34,\ I : 0.33]$, Q_2: $[G : 0.36,\ H : 0.32,\ I : 0.32]$, Q_3: $[G : 0.32,\ H : 0.35,\ I : 0.33]$, and Q_4: $[G : 0.33,\ H : 0.34,\ I : 0.33]$) where G, H, and I denote the native languages (cf. 2_{nd} column in Table 1). We can identify and remove incorrect predictions by using entropy as a measure of uncertainty in the prediction aggregation process as indicated in the *third* column of 1. Assume that the value of κ is equal to 50. As indicated by the symbol $*$ (with the low entropy values), the top 50% most certain predictions belong to Q_2 and Q_3. The average PMF of Q_2 and Q_3 is used to calculate the final prediction of the entire query document $\mathcal{Q}$.

Table 1. Prediction Aggregation Process (*Top most certain κ50% predictions).

Query Fragment (Q)	**Query Fragment Prediction (PMF)**	**Entropy**
Q_1	$[G : 0.33,\ H : 0.34,\ I : 0.33]$	1.5848
Q_2^*	$[G : 0.36,\ H : 0.32,\ I : 0.32]$	1.5827
Q_3^*	$[G : 0.32,\ H : 0.35,\ I : 0.33]$	1.5840
Q_4	$[G : 0.33,\ H : 0.34,\ I : 0.33]$	1.5848
Final Prediction	$[G : 0.34,\ H : 0.335,\ I : 0.325]$	–

5 Performance Evaluation

In this section, we describe the experimental setup, previous state-of-the-art approaches, and the results of our extensive experimental experiments.

5.1 Experimental Setup

In this subsection, we illustrate the statistics of the corpora used in this study, along with parameter settings, evaluation measures, and evaluation strategy.

Corpora. We created three corpora which are obtained from Project Gutenberg [71] written in French, German, and English, respectively. Each document is from a different author *to avoid any influence of the authorship identification task*. The Table 2 shows the details of each corpus.

Table 2. The three languages and their L1 classes are broken down.

English			**French**			**German**		
L1	**# Docs.**	**Text Length**	**L1**	**# Docs.**	**Text Length**	**L1**	**# Docs.**	**Text Length**
French	32	312,234	English	38	420,174	English	36	310,487
German	31	381,187	Dutch	23	431,158	French	24	304,647
Spanish	23	497,033	Finnish	19	371,540	Dutch	20	334,321
Swedish	7	417,475	Portuguese	14	422,152	Finnish	12	369,073
Norwegian	7	357,743	Russian	6	350,875	Portuguese	8	301,054

Evaluation Measures. As explained in Section 4 we make native-language predictions at two levels, namely, (i) fragment-level prediction; and (ii) document-level prediction. Thus, we evaluate accuracy at these two levels.

1. *Fragment Accuracy:* A fragment is correctly predicted if the correct L1 is identified as the query document's *most likely L1*.
2. *Document Accuracy:* The aggregated prediction (final prediction) produced from the prediction aggregation process with the correct L1 as the *most likely L1* of the query document.

Parameters Setting. Although not shown here, we experimented with different values for each parameter and found that the parameter values listed in Table 3 achieved the best accuracy. For PkNN, the k value denotes the k SSFs that are closest to the Q. The values, $(50\%, 100\%]$ denotes the MHD ranges while $(50\%, 75\%]$ PHD ranges, respectively. The L value denotes the size of the query fragment, Q, i.e., 15,000 tokens. The l value represents the chunk size, i.e., 1500 tokens. The κ number represents the percentage of query fragment predictions that are taken into account throughout the prediction aggregation procedure to produce the final prediction for a query document.

Table 3. Default parameter settings.

k	**MHD**	PHD	L	l	κ
10	(50%, 100%]	(50%, 75%]	15,000 tokens	1,500 tokens	50%

Parameter Settings for Pre-trained Models: To fine-tune pre-trained multilingual language models, we use the huggingface Tensorflow implementation for fine-tuning [72], with the hyper-parameters given in Table 4. All models are base models, with 12 layers (except DistilBERT, which has 6 layers), a hidden size of 768, and 12 attention heads. DistilBERT has a vocabulary size of 31K tokens, mBERT has a vocabulary size of 120K tokens, and Multilingual DeBERTa, and XLM-RoBERTa have a vocabulary size of 250K tokens.

Table 4. Parameter settings of the pre-trained multilingual language models.

Pre-trained Language Models	
Parameter	Value
# Epochs	5
Batch Size	8
Optimizer	Adam
Learning Rate	$2e^{-5}$
Loss	BinaryCrossentropy

Evaluation Strategy: For every experimental study, the reported results are average accuracy obtained through *leave-one-out* cross-validation. Recall that, similar to documents in the corpus,

we partition the query document into fragments and run an independent set similarity query for each query fragment. The *leave-one-out* validation assures that a document used for model testing is utilized exclusively for model testing.

5.2 Competitive Techniques

In this subsection, we provide the descriptions of the competitive techniques. Our solution (SS-NLI) is compared against the current state-of-the-art native language identification method (Comp-NLI) [3] and its enhanced version. Moreover, we also compare the performance of our solution (SS-NLI) against machine learning methods that have been extensively employed in recently published state-of-the-art studies to perform the native language identification task. These classifiers include *support vector machines (SVM)*, *linear discriminant analysis (LDA)* and *maximum entropy classifier* (MaxEnt) [3, 47, 62]. The descriptions of all competitors are provided in the following paragraphs.

Comp-NLI. This competitive technique is based on linear *support vector machines (SVM)* classifier and they used several types of features including POS uni-grams, POS bi-grams, POS tri-grams, and function words. SVM classifier creates statistical models to distinguish between the pairs of L1 classes in the training data by representing each document as a point in multidimensional space. To distinguish between L1 classes, the SVM model uses the mathematical means to build a decision boundary (or hyperplane) that best separates the documents of a specific L1 class from the documents of other L1 classes by maximizing the distance between the L1 classes and the hyperplane that separates them [3]. In contrast to Comp-NLI [3], each document in our representation is represented as a series of point sets.As a result, our solution is capable of (i) capturing language usage and production patterns of the authors, (ii) handling noise in the data; and (iii) removing uncertain predictions of a query document before combining them to produce a single prediction for the entire query document, which in turn help outperform the existing state-of-the-art NLI method.

Comp-NLI-E. Moreover, in this paper, we formulate an improved variant of the *Comp-NLI*. We found that, (i) partitioning each document into chunks. (ii) producing the probabilistic output for each chunk; and (iii) applying our prediction aggregation process illustrated in Section 4 improves the accuracy of Comp-NLI (see Section 5.3 for performance comparison). We call this improvement in the competitive technique, Comp-NLI *entropy*, which is abbreviated to Comp-NLI-E for conciseness.

In addition to this, we have compared the performance of our solution (SS-NLI) against machine learning methods that have been extensively employed in recently published state-of-the-art studies to perform native language identification tasks (see Section 5.3). These classifiers include *Linear discriminant analysis (LDA)* and *Maximum entropy classifier* (MaxEnt) [47, 62]. The descriptions of these machine learning methods are given in the following paragraphs.

LDA-NLI. *Linear discriminant analysis (LDA)* is similar to the SVM classifier in the sense that LDA also represents each document in the training data as a point in the multidimensional space. Instead of creating margins to separate L1 classes, the LDA determines the mathematical centroid of all the documents representing a particular L1 class. LDA thus determines a separate centroid for each L1 class and it classifies the test document by plotting it in the multidimensional space by computing its distance to each centroid. The test document is identified as belonging to an L1 class with minimum distance [62].

MaxEnt-NLI. *Maximum entropy classifier* (MaxEnt) relies on the concept of information theory which states that if nothing is known regarding a test document, the probability of its L1 class association is equally distributed across all the L1 classes until we start examining each feature of the document and how these features are distributed across the documents representing different L1 classes. For instance, in the case of eight candidate L1 classes and one of them is Norwegian, the prior probability that the writer of the test document is Norwegian is 1/8. Notwithstanding, if the test document contains a feature that occurs 25% of the time in the documents written by Norwegian authors, then the probability that the author of the test document is Norwegian increases to 1/4. It also modifies the other L1 classes in a way that is proportional to the distribution of that feature across different L1 classes. In case of multiple features, the MaxEnt model assigns the probability distribution as evenly as possible, so that it calculates the entropy of all the conditional probabilities and identifies the most unconstrained distribution [47, 62].

DistilBERT. DistilBERT is a streamlined version of the formidable BERT (Bidirectional Encoder Representations from Transformers) model, designed to deliver remarkable computational efficiency without compromising on performance. At its core, DistilBERT employs a clever knowledge distillation technique. It learns from the immense linguistic knowledge contained in BERT, a larger and more resource-intensive model, and distills this knowledge into a leaner form. One of the most

notable advantages of DistilBERT is its significantly improved inference speed. This makes it an excellent choice for real-time applications, including chatbots, virtual assistants, and recommendation systems, where rapid responses are crucial. Moreover, DistilBERT's versatility is evident in its ability to excel across a wide range of NLP tasks. Through pre-training on extensive text data followed by task-specific fine-tuning, it maintains competitive performance levels. This means it can handle tasks such as native language identification, sentiment analysis, text classification, and named entity recognition with precision. Accessing DistilBERT is a breeze, thanks to the Hugging Face Transformers library, making it a valuable tool for developers and researchers seeking to leverage its efficiency and capability for their NLP endeavors. In essence, DistilBERT represents a pivotal step forward in the pursuit of efficient, high-performance NLP models, unlocking a world of possibilities in language understanding and generation.

mBERT. Multilingual BERT, commonly known as mBERT, is a pioneering language model developed by Google AI. It stands out in the field of Natural Language Processing (NLP) for its ability to understand and process text in multiple languages. mBERT is an extension of the original BERT (Bidirectional Encoder Representations from Transformers) model, specifically tailored to address the challenges posed by multilingual text. At its core, mBERT is a pre-trained model that captures contextual information from text in over 100 different languages. It learns to understand nuances, idiomatic expressions, and grammatical structures across these diverse languages, making it a valuable resource for a wide range of multilingual NLP tasks. One of the most significant advantages of mBERT is its versatility. Researchers and developers can leverage this model for various tasks, including native language identification, sentiment analysis, machine translation, and cross-lingual information retrieval. By fine-tuning mBERT on specific tasks, it can adapt its multilingual knowledge to perform exceptionally well in a targeted application. mBERT has democratized access to NLP capabilities across languages and regions, enabling applications that cater to a global audience. Its availability and flexibility make it an indispensable tool for those seeking to bridge language barriers and develop multilingual NLP solutions. In essence, mBERT paves the way for a more connected and linguistically inclusive digital landscape.

DeBERTa. DeBERTa, or Decoding-enhanced BERT with Disentangled Attention, is an advanced language model that pushes the boundaries of language understanding and generation. Developed as an evolution of the BERT (Bidirectional Encoder Representations from Transformers) architecture, DeBERTa introduces several groundbreaking features that enhance its capabilities. The most notable innovation in DeBERTa is its disentangled attention mechanism, which allows the model to focus on specific parts of a text while ignoring irrelevant information. This fine-grained control over attention makes DeBERTa exceptionally effective at capturing subtle nuances and long-range dependencies in language, making it a versatile choice for a wide range of NLP tasks. With its advanced architecture, DeBERTa achieves state-of-the-art results across various NLP benchmarks and tasks, including text classification, question-answering, and language modeling. Researchers and developers can fine-tune DeBERTa on specific tasks to harness its remarkable language understanding and generation capabilities. DeBERTa exemplifies the continuous innovation in NLP, offering enhanced performance, fine-grained attention control, and improved text generation. Its availability empowers the development of more sophisticated and context-aware NLP applications, further enriching the capabilities of natural language understanding and generation systems.

XLMRoBERTa. The XLM-RoBERTa, an exceptional model in the realm of Natural Language Processing (NLP), represents the fusion of two powerful language models: RoBERTa and XLM. Developed by Facebook AI, XLM-RoBERTa stands out for its remarkable ability to understand and process text in multiple languages, making it a cornerstone in the field of multilingual NLP. At its core, XLM-RoBERTa combines the strengths of RoBERTa, which is known for its state-of-the-art performance in monolingual tasks, with XLM, a model designed for cross-lingual applications. This fusion results in a versatile model capable of handling a broad spectrum of languages and linguistic complexities. XLM-RoBERTa's pre-training involves learning from a diverse corpus of text in over 100 languages, allowing it to capture linguistic nuances, idiomatic expressions, and grammatical structures across a multitude of languages. This comprehensive understanding is invaluable for various multilingual NLP tasks, including machine translation, language identification, sentiment analysis, and cross-lingual information retrieval. We can fine-tune XLM-RoBERTa for specific tasks, ensuring that it adapts its multilingual knowledge effectively. The model's performance consistently ranks at the forefront of multilingual benchmarks, highlighting its efficacy and versatility in enabling global communication and information access. In essence, XLM-RoBERTa is a game-changer for multilingual NLP, providing a powerful solution for bridging language barriers, fostering linguistic inclusivity, and enhancing cross-cultural communication in a digitally connected world.

5.3 Experimental Results

We conducted four experimental studies in the context of this investigation's main objectives and contributions, which are listed in Section Introduction. The first experimental study aims at feature evaluation. That is, we investigate the effectiveness of our document representation model to capture the overlapping information between feature spaces. Specifically, we compare the performances of three feature sets, namely, (i) POS bi-grams; (ii) POS tri-grams; and (iii) the combination of POS bi-grams and POS tri-grams (Section 5.3). In a second experimental study, we evaluate the outlier handling mechanisms associated with set distance measures. Specifically, We compare the accuracy of three different set distance measures utilized in this study: (i) *standard Hausdorff distance (SHD*), (ii) *partial Hausdorff distance (PHD)* and (iii) *modified Hausdorff distance (MHD)*. In the third experimental study, we compare the performance of our solution to that of competitors. We specifically compared the performance of our solution to the existing state-of-the-art competitive technique presented in multilingual native language identification (Comp-NLI), its improved variant (Comop-NLI-E) [3], and two classical machine learning techniques widely used in previous studies, namely, LDA-NLI and MaxEnt-NLI (Section 5.3). In the fourth experimental study, we compare the performance of L2s about their L1. In terms of the L1 classes of the three languages, we compare the performance of our solution (SS-NLI) to an improved variation of the competitive method (Comp-NLI-E) (Section 5.3).

Features Evaluation: Proposed Method In this study, we assessed the performance of our basic feature sets, namely, POS bi-grams and POS tri-grams. In addition, to reveal the overlapping information captured by our document representation model (i.e., collection of point sets), we combined POS bi-grams with POS tri-grams features into one vector and compared its performance against individual feature sets (i.e., POS bi-grams and POS tri-grams). We evaluated the performance of our feature sets across three languages. For each feature set as shown in Table 5, the performance of the proposed solution was approximately similar *across different languages*. In addition, the *combined feature set* had outperformed the individual feature sets across three languages. These findings imply that some of the stylometric data captured by our features was complementary and orthogonal. The performance of NLI was improved as a result of integrating these features. We will limit the rest of the experimental investigations to the *combined feature set* and report only document accuracy because the *combined feature set* outperforms individual feature sets across the three languages and document accuracy is always higher than fragment accuracy.

Table 5. Proposed Solution: Fragment accuracy (%) & Document Accuracy (%) comparison between our feature sets across the three languages using modified Hausdorff distance (MHD).

Feature Sets	**English**	**French**	**German**
POS Bi-grams	61.31 & 66.0	61.77 & 67.0	63.93 & 70.0
POS Tri-grams	64.24 & 71.0	56.01 & 70.0	65.04 & 72.0
POS Bi-grams & POS Tri-grams	75.41 & 89.0	77.16 & 89.0	74.93 & 88.0

Accuracy Comparison Among Set Distance Measures: Proposed Method In this section, we compare the accuracy of three different set distance measures used in this study including (i) *standard Hausdorff distance (SHD*), (ii) *partial Hausdorff distance (PHD)* and (iii) *modified Hausdorff distance (MHD)*. Experimental results given in Table 6 shows that MHD outperforms two other set distance measures. Recall that SHD does not have an outlier handling mechanism, whereas MHD and PHD have outlier handling mechanisms (please see Section 4.2 for details). The fact that MHD outperforms SHD indicates that our dataset contains noise. As evidenced by its superior performance, MHD has a better outlier handling mechanism than PHD.

Performance Comparison between Our Solution and Classical Machine Learning Based Competitive Techniques In this section, we assessed the performance of the proposed solution against the existing state-of-the-art competitive technique presented in multilingual native language identification (Comp-NLI) [3], its improved variations (Comop-NLI-E) and two classical machine learning techniques extensively used in previous studies and the existing state-of-the-art

Table 6. Proposed Solution (Document Accuracy): Accuracy comparison among set similarity measures based on POS Bi-grams & POS Tri-grams.

Set Similarity Measure	English	French	German
SHD	74.0	75.0	74.0
PHD	78.0	80.0	79.0
MHD	89.0	89.0	88.0

language models including DistilBERT [14], mBERT [15], multilingual DeBERTa [16], and XLM-RoBERTa [17]. We used the same evaluation strategy as illustrated in section 5.1. The experimental results are given in Table 7. As can be seen, the proposed solution (SS-NLI) outperforms the Comp-NLI, its improved variation, comp-NLI-E, other classical machine learning methods, and pre-trained language models. Besides that, the performance of the proposed solution and competitive techniques is approximately similar across the three languages. However, this is not the case with pre-trained language models. That is, fine-tuning pre-trained language models provides better performance in English than in French or German.

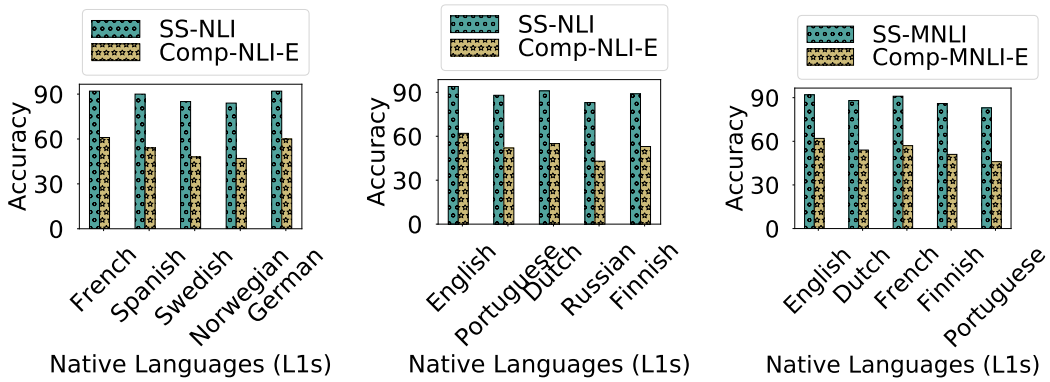
Table 7. Document Accuracy: Performance comparison between proposed solution and competitive techniques based on POS Bi-grams & POS Tri-grams using MHD.

Methods	English	French	German
MaxEnt-NLI	40.0	39.0	38.0
LDA-NLI	44.0	42.0	43.0
Comp-NLI	53.0	49.0	52.0
Comp-NLI-E	58.0	57.0	58.0
DistilBERT	88.0	83.0	85.0
mBERT	87.0	86.0	87.0
DeBERTa	88.0	86.0	83.0
XLM-RoBERTa	88.0	85.0	84.0
SS-NLI	**89.0**	**89.0**	**88.0**

Performance Comparison among L2's in Terms of their L1. In this study, we provide the performance comparison between the proposed solution (SS-NLI) and the improved variation of competitive technique (Comp-NLI-E) for the three languages in terms of their L1 classes. As shown in Figure 3 the proposed solution (SS-NLI) outperformed the improved variation of the competitive technique (Comp-NLI-E) This is because our document representation model (i) is capable of capturing language usage and production patterns both within and across the documents, (ii) is capable of handling the outliers in our data, and (iii) allows us to remove uncertain predictions from a query document before producing the final predictions. This document representation model, in particular, allows us to use set similarity functions with outlier handling mechanisms (e.g., *modified Hausdorff distance*), which improves the performance of the NLI task. Furthermore, the performance of the proposed solution in terms of L1 classes appears to be similar in all three languages, as shown in Figure 3 (Section 5.3 contains a detailed comparison of the proposed solution and all competing techniques).

6 Conclusions

This paper presents an NLI solution. To conduct this research, we created three corpora written in French, German, and English. We then propose a solution that captures the language usage and production patterns of the authors in documents to perform the NLI task. With the help of extensive experimental studies, we show that our solution can be applied to different languages and reports higher accuracy than state-of-the-art NLI methods. We intend to apply our solution to additional languages in the future, assuming that the relevant corpora are available. In the future, provided the relevant corpora, we plan to apply our solution to additional languages. Furthermore, the integration of cultural references and materials, in addition to linguistic aspects, could help to personalize further and enrich educational content. Exploring the possibilities of direct integration with educational platforms and tools, such as learning management systems (LMS) or adaptive



(a) English (L2) (b) French (L2) (c) German (L2)

Fig. 3. Document Accuracy: Performance comparison of the proposed solution (SS-NLI) and the competitive technique (Comp-NLI-E) in terms of L1 classes for the three languages MHD POS Bi-grams & POS Tri-grams.

learning technologies, could represent a promising field of research for the future. Such integrations could facilitate the practical application of the results of this paper, making them more accessible to both educators and students.

References

[1] Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel R. Tetreault. *Automated Grammatical Error Detection for Language Learners*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2010.

[2] Moshe Koppel, Jonathan Schler, and Kfir Zigdon. "Determining an author's native language by mining a text for errors". In: *Proceedings of the Eleventh ACM SIGKDD, Chicago, Illinois, USA, August 21-24, 2005*. 2005, pp. 624–628.

[3] Shervin Malmasi and Mark Dras. "Multilingual native language identification". In: *Natural Language Engineering* 23.2 (2017), pp. 163–215.

[4] Raheem Sarwar, Qing Li, Thanawin Rakthanmanon, and Sarana Nutanong. "A scalable framework for cross-lingual authorship identification". In: *Information Sciences* 465 (2018), pp. 323–339.

[5] Raheem Sarwar, Chenyun Yu, Ninad Tungare, Kanatip Chitavisutthivong, Sukrit Sriratanawilai, Yaohai Xu, Dickson Chow, Thanawin Rakthanmanon, and Sarana Nutanong. "An Effective and Scalable Framework for Authorship Attribution Query Processing". In: *IEEE Access* 6 (2018), pp. 50030–50048.

[6] Magdalena Saldana-Perez, Marco Moreno-Ibarra, Carolina Palma-Preciado, Giovanni Guzman, and Yanil Contreras-Jimenez. "Reinforcing Tourism Post-pandemic Through a Natural Language Processing Data Analysis". In: *The International Research & Innovation Forum*. Springer. 2023, pp. 591–605.

[7] Aleksandra Pawlicka, Marek Pawlicki, Rafał Kozik, Wiktor Kurek, and Michał Choraś. "How Explainable Is Explainability? Towards Better Metrics for Explainable AI". In: *The International Research & Innovation Forum*. Springer. 2023, pp. 685–695.

[8] Bokolo Anthony Jnr, Sobah Abbas Petersen, Markus Helfert, and Hong Guo. "Digital transformation with enterprise architecture for smarter cities: a qualitative research approach". In: *Digital policy, regulation and governance* 23.4 (2021), pp. 355–376.

[9] Fedelucio Narducci, Pierpaolo Basile, Cataldo Musto, Pasquale Lops, Annalina Caputo, Marco de Gemmis, Leo Iaquinta, and Giovanni Semeraro. "Concept-based item representations for a cross-lingual content-based recommendation process". In: *Information Sciences* 374 (2016), pp. 15–31.

[10] Daniel P. Huttenlocher, Gregory A. Klanderman, and William Rucklidge. "Comparing images using the Hausdorff distance". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 15.9 (1993), pp. 850–863.

[11] Marie-Pierre Dubuisson and Anil K. Jain. "A modified Hausdorff distance for object matching". In: *12th IAPR International Conference on Pattern Recognition, Conference A: Computer*

Vision & Image Processing, ICPR 1994, Jerusalem, Israel, 9-13 October, 1994. 1994, pp. 566–568.

[12] CC Holmes and NM Adams. "A probabilistic nearest neighbour method for statistical pattern recognition". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 64.2 (2002), pp. 295–306.

[13] Anne Abeillé, Lionel Clément, and François Toussenel. "Building a treebank for French". In: *Treebanks* (2003), pp. 165–187.

[14] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: *ArXiv* abs/1910.01108 (2019).

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186.

[16] Pengcheng He, Jianfeng Gao, and Weizhu Chen. "DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing". In: *ArXiv* (2021). arXiv: 2111.09543 [cs.CL].

[17] Alexis Conneau et al. "Unsupervised Cross-lingual Representation Learning at Scale". In: 2020, pp. 8440–8451.

[18] Syed Konain Abbas, Muhammad Usman Ghani Khan, Jia Zhu, Raheem Sarwar, Naif R Aljohani, Ibrahim A Hameed, and Muhammad Umair Hassan. "Vision based intelligent traffic light management system using Faster R-CNN". In: *CAAI Transactions on Intelligence Technology* (2024).

[19] Kanishka Silva, Ingo Frommholz, Burcu Can, Fred Blain, Raheem Sarwar, and Laura Ugolini. "Forged-GAN-BERT: Authorship Attribution for LLM-Generated Forged Novels". In: *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*. 2024, pp. 325–337.

[20] Muhammad Umair Hassan, Xiuyang Zhao, Raheem Sarwar, Naif R Aljohani, and Ibrahim A Hameed. "SODRet: Instance retrieval using salient object detection for self-service shopping". In: *Machine Learning with Applications* 15 (2024), p. 100523.

[21] Raheem Sarwar, Maneesha Perera, Pin Shen Teh, Raheel Nawaz, and Muhammad Umair Hassan. "Crossing Linguistic Barriers: Authorship Attribution in Sinhala Texts". In: *ACM Transactions on Asian and Low-Resource Language Information Processing* (2024).

[22] Kanishka Silva, Burcu Can, Raheem Sarwar, Frederic Blain, and Ruslan Mitkov. "Text data augmentation using generative adversarial networks–a systematic review". In: *Journal of Computational and Applied Linguistics* 1 (2023), pp. 6–38.

[23] Kanishka Silva, Burcu Can, Frédéric Blain, Raheem Sarwar, Laura Ugolini, and Ruslan Mitkov. "Authorship attribution of late 19th century novels using GAN-BERT". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*. 2023, pp. 310–320.

[24] Hadeel Saadany, Emad Mohamed, and Raheem Sarwar. "Towards a better understanding of Tarajem: creating topological networks for Arabic biographical dictionaries". In: *Journal of Data Mining & Digital Humanities* (2023).

[25] Emad Mohamed, Raheem Sarwar, and Sayed Mostafa. "Translator attribution for Arabic using machine learning". In: *Digital Scholarship in the Humanities* 38.2 (2023), pp. 658–666.

[26] Raheem Sarwar. "Author Gender Identification for Urdu Articles". In: *International Conference on Computational and Corpus-Based Phraseology*. Springer. 2022, pp. 221–235.

[27] Emad Mohamed and Raheem Sarwar. "Linguistic features evaluation for hadith authenticity through automatic machine learning". In: *Digital Scholarship in the Humanities* 37.3 (2022), pp. 830–843.

[28] Nattapol Trijakwanich, Peerat Limkonchotiwat, Raheem Sarwar, Wannaphong Phatthiyaphaibun, Ekapol Chuangsuwanich, and Sarana Nutanong. "Robust fragment-based framework for cross-lingual sentence retrieval". In: Association for Computational Linguistics. 2021.

[29] Iqra Safder et al. "Sentiment analysis for Urdu online reviews using deep learning models". In: *Expert Systems* 38.8 (2021), e12751.

[30] Peerat Limkonchotiwat, Wannaphong Phatthiyaphaibun, Raheem Sarwar, Ekapol Chuangsuwanich, and Sarana Nutanong. "Domain adaptation of Thai word segmentation models using stacked ensemble". In: Association for Computational Linguistics. 2020.

[31] Saeed-Ul Hassan, Naif R Aljohani, Mudassir Shabbir, Umair Ali, Sehrish Iqbal, Raheem Sarwar, Eugenio Martínez-Cámara, Sebastián Ventura, and Francisco Herrera. "Tweet coupling: A

social media methodology for clustering scientific publications". In: *Scientometrics* 124 (2020), pp. 973–991.

[32] Raheem Sarwar, Attapol T Rutherford, Saeed-Ul Hassan, Thanawin Rakthanmanon, and Sarana Nutanong. "Native language identification of fluent and advanced non-native writers". In: *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* 19.4 (2020), pp. 1–19.

[33] Raheem Sarwar, Norawit Urailertprasert, Nattapol Vannaboot, Chenyun Yu, Thanawin Rakthanmanon, Ekapol Chuangsuwanich, and Sarana Nutanong. "*CAG*: Stylometric authorship attribution of multi-author documents using a co-authorship graph". In: *IEEE Access* 8 (2020), pp. 18374–18393.

[34] Farah Adeeba and Sarmad Hussain. "Native Language Identification in Very Short Utterances Using Bidirectional Long Short-Term Memory Network". In: *IEEE Access* 7 (2019), pp. 17098–17110.

[35] Neelakshi Sarma, Sanasam Ranbir Singh, and Diganta Goswami. "Influence of social conversational features on language identification in highly multilingual online conversations". In: *Information Processing & Management* 56.1 (2019), pp. 151–166.

[36] Mirco Kocher and Jacques Savoy. "Distance measures in author profiling". In: *Information Processing & Management* 53.5 (2017), pp. 1103–1119.

[37] Francisco M. Rangel Pardo and Paolo Rosso. "On the impact of emotions on author profiling". In: *Information Processing & Management* 52.1 (2016), pp. 73–92.

[38] Iqra Ameer, Grigori Sidorov, and Rao Muhammad Adeel Nawab. "Author profiling for age and gender using combinations of features of various types". In: *Journal of Intelligent and Fuzzy Systems* 36.5 (2019), pp. 4833–4843.

[39] Gili Goldin, Ella Rabinovich, and Shuly Wintner. "Native Language Identification with User Generated Content". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. 2018, pp. 3591–3601.

[40] Lingzhen Chen, Carlo Strapparava, and Vivi Nastase. "Improving Native Language Identification by Using Spelling Errors". In: *Proceedings of the 55th Annual Meeting of the ACL*. 2017, pp. 542–546.

[41] Seifeddine Mechti, Ayoub Abbassi, Lamia Hadrich Belguith, and Rim Faiz. "An empirical method using features combination for Arabic native language identification". In: *13th IEEE/ACS International Conference of Computer Systems and Applications*. 2016, pp. 1–5.

[42] Avni Rajpal, Tanvina B. Patel, Hardik B. Sailor, Maulik C. Madhavi, Hemant A. Patil, and Hiroya Fujisaki. "Native Language Identification Using Spectral and Source-Based Features". In: *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*. 2016, pp. 2383–2387.

[43] Laura Mayfield Tomokiyo and Rosie Jones. "You're Not From 'Round Here, Are You? Naive Bayes Detection of Non-Native Utterances". In: *Language Technologies 2001: The Second Meeting of the NAACL*. 2001.

[44] Andrzej Kulig, Jaroslaw Kwapien, Tomasz Stanisz, and Stanislaw Drozdz. "In narrative texts punctuation marks obey the same statistics as words". In: *Information Sciences* 375 (2017), pp. 98–113.

[45] Julian Brooke and Graeme Hirst. "Native language detection with 'cheap'learner corpora". In: *Twenty Years of Learner Corpus Research. Looking Back, Moving Ahead*. Presses universitaires de Louvain. 2013, pp. 1–37.

[46] Stanislaw Drozdz, Pawel Oswiecimka, Andrzej Kulig, Jaroslaw Kwapien, Katarzyna Bazarnik, Iwona Grabska-Gradzinska, Jan Rybicki, and Marek Stanuszek. "Quantifying origin and character of long-range correlations in narrative texts". In: *Information Sciences* 331 (2016), pp. 32–44.

[47] Sze-Meng Jojo Wong, Mark Dras, and Mark Johnson. "Exploring adaptor grammars for native language identification". In: *Proceedings of the 2012 Joint Conference on EMNLP and CoNLL*. 2012, pp. 699–709.

[48] Dominique Estival, Tanja Gaustad, Son Bao Pham, Will Radford, and Ben Hutchinson. "Author profiling for English emails". In: *Proceedings of the 10th Conference of the PACLING*. 2007, pp. 263–272.

[49] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. "Supervised machine learning: A review of classification techniques". In: *Emerging artificial intelligence applications in computer engineering* 160 (2007), pp. 3–24.

[50] Joel R Tetreault, Daniel Blanchard, and Aoife Cahill. "A Report on the First Native Language Identification Shared Task." In: *BEA@ NAACL-HLT*. 2013, pp. 48–57.

[51] Joel Tetreault, Daniel Blanchard, Aoife Cahill, and Martin Chodorow. "Native tongues, lost and found: Resources and empirical evaluations in native language identification". In: *Proceedings of COLING 2012* (2012), pp. 2585–2602.

[52] Raheem Sarwar, Afifa Zia, Raheel Nawaz, Ayman Fayoumi, Naif Radi Aljohani, and Saeed-Ul Hassan. "Webometrics: evolution of social media presence of universities". In: *Scientometrics* 126 (2021), pp. 951–967.

[53] Iqra Safder, Hafsa Batool, Raheem Sarwar, Farooq Zaman, Naif Radi Aljohani, Raheel Nawaz, Mohamed Gaber, and Saeed-Ul Hassan. "Parsing AUC result-figures in machine learning specific scholarly documents for semantically-enriched summarization". In: *Applied Artificial Intelligence* 36.1 (2022), p. 2004347.

[54] Fahad Sabah, Yuwen Chen, Zhen Yang, Muhammad Azam, Nadeem Ahmad, and Raheem Sarwar. "Model optimization techniques in personalized federated learning: A survey". In: *Expert Systems with Applications* (2023), p. 122874.

[55] Saeed-Ul Hassan, Naif Radi Aljohani, Usman Iqbal Tarar, Iqra Safder, Raheem Sarwar, Salem Alelyani, and Raheel Nawaz. "Exploiting tweet sentiments in altmetrics large-scale data". In: *Journal of Information Science* 49.5 (2023), pp. 1229–1245.

[56] Muhammad Umair Hassan, Saleh Alaliyat, Raheem Sarwar, Raheel Nawaz, and Ibrahim A Hameed. "Leveraging deep learning and big data to enhance computing curriculum for industry-relevant skills: A Norwegian case study". In: *Heliyon* 9.4 (2023).

[57] Raheem Sarwar and Saeed-Ul Hassan. "Urduai: Writeprints for Urdu authorship identification". In: *Transactions on Asian and Low-Resource Language Information Processing* 21.2 (2021), pp. 1–18.

[58] Muhammad Kashif Afzal et al. "Generative image captioning in Urdu using deep learning". In: *Journal of Ambient Intelligence and Humanized Computing* (2023), pp. 1–13.

[59] Abu Bakar, Raheem Sarwar, Saeed-Ul Hassan, and Raheel Nawaz. "Extracting Algorithmic Complexity in Scientific Literature for Advance Searching". In: *Journal of Computational and Applied Linguistics* 1 (2023), pp. 39–65.

[60] Fahad Sabah, Yuwen Chen, Zhen Yang, Abdul Raheem, Muhammad Azam, and Raheem Sarwar. "Heart Disease Prediction with 100% Accuracy, Using Machine Learning: Performance Improvement with Features Selection and Sampling". In: *2023 8th IEEE International Conference on Network Intelligence and Digital Content (IC-NIDC)*. IEEE. 2023, pp. 41–45.

[61] Dimah Alahmadi, Amal Babour, Kawther Saeedi, and Anna Visvizi. "Ensuring inclusion and diversity in research and research output: A case for a language-sensitive NLP crowdsourcing platform". In: *Applied Sciences* 10.18 (2020), p. 6216.

[62] Scott Jarvis and Magali Paquot. *Native language identification*. Cambridge University Press, 2015.

[63] Raheem Sarwar, Thanasarn Porthaveepong, Attapol Rutherford, Thanawin Rakthanmanon, and Sarana Nutanong. "StyloThai: A Scalable Framework for Stylometric Authorship Identification of Thai Documents". In: *ACM Trans. Asian & Low-Resource Lang. Inf. Process.* 19.3 (2019), 36:1–36:15.

[64] Raheem Sarwar, Pin Shen Teh, Fahad Sabah, Raheel Nawaz, Ibrahim A Hameed, Muhammad Umair Hassan, et al. "AGI-P: A Gender Identification Framework for Authorship Analysis Using Customized Fine-Tuning of Multilingual Language Model". In: *IEEE Access* (2024).

[65] Raheel Nawaz, Ernest Edem Edifor, Samantha Reive Holland, Qi Cao, and Leo Shixiong Liu. "The impact of degree apprenticeships: analysis, insights and policy recommendations". In: *Transforming Government: People, Process and Policy* 17.3 (2023), pp. 372–386.

[66] *Foreign language*. http://ec.europa.eu/eurostat/statistics-explained/index.php/Foreign_language. (Accessed on 09/30/2023).

[67] Paul Thompson, Raheel Nawaz, Ioannis Korkontzelos, William Black, John McNaught, and Sophia Ananiadou. "News search using discourse analytics". In: *2013 Digital Heritage International Congress (DigitalHeritage)*. Vol. 1. IEEE. 2013, pp. 597–604.

[68] Slav Petrov, Dipanjan Das, and Ryan McDonald. "A universal part-of-speech tagset". In: *arXiv preprint arXiv:1104.2086* (2011).

[69] Rajalida Lipikorn, Akinobu Shimizu, and Hidefumi Kobatake. "A modified Hausdorff distance for object matching". In: *Pattern Recognition*. Vol. 1. 1994, pp. 566–568.

[70] Sarana Nutanong, Chenyun Yu, Raheem Sarwar, Peter Xu, and Dickson Chow. "A Scalable Framework for Stylometric Analysis Query Processing". In: *IEEE 16th International Conference on Data Mining (ICDM)*. IEEE. 2016, pp. 1125–1130.

[71] *Free eBooks | Project Gutenberg*. https://www.gutenberg.org/. (Accessed on 09/30/2023).

[72] *Fine-tune a pretrained model*. https://huggingface.co/docs/transformers/training. (Accessed on 09/30/2023).