

Artificial Intelligence Methods for Spanish Documents Classification

Métodos de inteligencia artificial
para la clasificación de documentos en español

Métodos de inteligência artificial
para a classificação de documentos em espanhol

Tad Gonsalves

Sophia University, Tokyo, Japan

t-gonsal@sophia.ac.jp

<https://orcid.org/0000-0001-9424-3078>

Hang Hu

Sophia University, Tokyo, Japan

h-hu-5w8@eagle.sophia.ac.jp

<https://orcid.org/0009-0007-9008-7494>

Yoshimi Hiroyasu

Sophia University, Tokyo, Japan

y-hiroya@sophia.ac.jp

<https://orcid.org/0000-0001-5596-9933>

Abstract

The rapid globalization and growing need for cross-language communication necessitate modern, real-time corpora to aid language learners. Traditional methods for creating such corpora, especially in Spanish, are inadequate due to their inability to process the vast and unstructured data available online. This study explores Artificial Intelligence (AI) methodologies for automatic Spanish document acquisition from the web, pre-processing and classifying them in order to build a vast and flexible corpus for Spanish learning. The research applies web crawling using the Scrapy framework to collect data, which is then cleaned and classified using advanced Natural Language Processing (NLP) models. Specifically, the study employs BERT (Bidirectional Encoder Representations from Transformers) and its enhanced variant RoBERTa to achieve document classification. Through a combination of data augmentation techniques and deep learning models, the study achieves high accuracy in classifying Spanish-language texts, demonstrating the potential for using AI to overcome the limitations of traditional corpus-building approaches.

Keywords: artificial intelligence; machine learning; deep learning; data augmentation; document classification.

Resumen

La rápida globalización y la creciente necesidad de comunicación interlingüística requieren corpus modernos y en tiempo real para ayudar a los estudiantes de idiomas. Los métodos tradicionales para crear dichos corpus, especialmente en español, son inadecuados debido a su incapacidad para procesar la gran cantidad de datos no estructurados disponibles en internet. En este estudio se exploran las metodologías de inteligencia artificial (IA) para la adquisición automática de documentos en español de la web, preprocesándolos y clasificándolos con el fin de construir un corpus vasto y flexible para el aprendizaje del español. La investigación aplica el rastreo web mediante el framework Scrapy para recopilar datos, que luego se limpian y clasifican utilizando modelos avanzados de procesamiento del lenguaje natural (PLN). En concreto, el estudio emplea el algoritmo BERT (Bidirectional Encoder Representations from Transformers) y su variante mejorada RoBERTa para lograr la clasificación de documentos. Mediante una combinación de técnicas de aumento de datos y modelos de aprendizaje profundo, el estudio logra una alta precisión en la clasificación de texto en español, lo que demuestra el potencial del uso de la IA para superar las limitaciones de los enfoques tradicionales de creación de corpus.

Palabras clave: inteligencia artificial; aprendizaje automático; aprendizaje profundo; aumento de datos; clasificación de documentos.

Resumo

A rápida globalização e a crescente necessidade de comunicação entre línguas exigem corpora modernos e em tempo real para ajudar os estudantes de línguas. Os métodos tradicionais para criar tais corpora, especialmente em espanhol, são inadequados devido à sua incapacidade de processar os dados vastos e não estruturados disponíveis online. Este estudo explora metodologias de Inteligência Artificial (IA) para a aquisição automática de documentos espanhóis da Web, pré-processando-os e classificando-os de modo a construir um corpus vasto e flexível para a aprendizagem do espanhol. A investigação aplica o rastreo da Web utilizando a estrutura Scrapy para recolher dados, que são depois limpos e classificados utilizando modelos avançados de processamento da linguagem natural (PNL). Especificamente, o estudo utiliza o algoritmo BERT (Bidirectional Encoder Representations from Transformers) e a sua variante melhorada RoBERTa para obter a classificação dos documentos. Através de uma combinação de técnicas de aumento de dados e modelos de aprendizagem profunda, o estudo consegue uma elevada precisão na classificação de textos em espanhol, demonstrando o potencial da utilização da IA para ultrapassar as limitações das abordagens tradicionais de construção de corpus.

Palavras-chave: inteligência artificial; aprendizagem automática; aprendizagem profunda; aumento de dados; classificação de documentos.

Received: 10/14/2024

Accepted: 11/13/2024

Published: 12/30/2024

1. Introduction

With the rapid rise of globalization, cross-language communication has become increasingly essential, leading to a growing interest in language learning. However, acquiring new languages remains challenging, particularly, for beginners and intermediate learners. These learners often struggle to find appropriate learning materials independently, because traditional textbooks frequently fail to reflect the dynamic, real-world language use, limiting students' progress.

To support language learners more effectively, it is crucial to provide updated real-time corpora that reflect modern usage. While the development of large corpora for various purposes has gained traction, most existing corpora are predominantly in English, as it dominates Natural Language Processing (NLP) research. Additionally, many corpora are unlabeled or contain duplicates, requiring manual or machine learning-based processing. This problem is even more pronounced for Spanish, where comprehensive labeled corpora are lacking.

Artificial Intelligence (AI) is greatly impacting language learning (Muñoz-Basols y Fuertes, 2024; Muñoz-Basols et al., 2023; Jerusha and Rajakumari, 2024). As part of the university level Spanish e-learning support project, this study deals with the development of the latest AI tools to address the

following three major computational steps in creating a Spanish corpus: 1) Documents acquisition, 2) Documents pre-processing, and 3) Document classification.

Documents acquisition refers to the technique of searching for and automatically retrieving relevant articles (documents) online. The world wide web offers vast amounts of multilingual text covering various domains and continually updated in real time, making it a valuable resource for constructing learning corpora. However, extracting meaningful content from the web is not straightforward. The web is flooded with heterogeneous, redundant and irrelevant information, making it challenging to filter and process data accurately. The AI tool rapidly crawls through a large amount of relevant Spanish web documents in the public domain and downloads them automatically through a software process known as web-scraping.

Documents pre-processing refers to the preliminary stage of removing unnecessary text bits from the acquired documents before performing the main NLP task, which in this case is document classification. Since most of the downloaded contents contain noise and other irrelevant features, they need to be fed into pre-processing pipelines for cleansing and filtering. The data cleaning process mainly involves removing irrelevant symbols and marks in the text. These characters often appear in the obtained network data, including Unicode identifiers, emoji representations, and special symbols. However, unlike the general pre-processing, the punctuation marks are retained in the original documents, since they are essential for preserving the semantics. Since the downloaded documents datasets are not sufficient for producing outstanding classification results, data augmentation is also performed at the pre-processing stage.

Document classification is the main NLP task. After pre-processing and filtering, the documents are fed into a deep learning model that learns to classify the documents in the dataset according to the pre-assigned labels (categories). The model is supposed to learn a general understanding of the contents of each document and thereby classify the documents as belonging to specific categories. Document classification is a fundamental task in NLP, which apart from corpora creation as in this study, has wide applications as in spam filtering (Abayomi-Alli et al., 2019), sentiment analysis (Medhat et al., 2014), fake news detection (Ahmed et al., 2018), customer reviews (Xu et al., 2011), anomalies in social media (Yang et al., 2015), and so on.

Statistical methods have long been employed in lexical analysis, text classification and other NLP tasks. However, their mathematical rigidity makes them suitable only for small-sized data structured in standard data formats. They cannot be easily scaled to handle web data which is massive, semi-structured, heterogeneous, redundant and dynamic in nature. Machine learning models greatly overcome the limitations of statistical methods; however, they are not suitable for modern large and complex data.

This study delves deep in the area of Artificial Intelligence (AI) to handle the large-scale and complex problem of text data acquisition, cleansing and classifying, addressing the challenges of building and maintaining a versatile Spanish corpus. It focuses on the acquisition, pre-processing and classification of two large publicly available news datasets. It uses BERT and its improved version, RoBERTa to achieve the task of document classification. The proposed AI methods yield over 97% accuracy in training and over 93% in testing.

2. Computational Methods for Text Classification

To build any useful corpus, it is necessary to first cleanse and classify the raw text data. Text classification is a fundamental task in text processing, and its applications are also pervasive, such as spam filtering, news classification, part-of-speech tagging, sentiment analysis, fake news detection and so on. It is not fundamentally different from other classifications. The core method is to classify by extracting the features of content data and then selecting the best matching category. The objects to be classified include short sentences, titles, comments, and long articles. According to the number of classification categories, the classification model can be divided into binary classification and multi-classification. Generally, the more the number of classification categories, the higher the requirements for the classifier's feature extraction ability and classification accuracy. The following sub-section introduces the various computational techniques used in text classification.

2.1. Statistical Methods

The task of text classification is a classic problem in natural language processing. The earliest related research can be traced back to classification based on Expert Systems. This classification requires the participation of domain experts, consumes a lot of human and material resources; besides, it is not very effective in coverage and accuracy.

Geometry-based models use a vector space to represent text as a multi-dimensional vector, viewed as a point in multi-dimensional space. These models use geometric principles to construct a hyperplane to separate the representation spaces of different texts. A commonly used algorithm is Support Vector Machine (SVM) (Cervantes et al., 2020). In the case of limited samples, the algorithm obtains the optimal solution in the sample space. This method maps the original linear indivisible space to a high-dimensional linear separable space. SVM constructs a linear function in a high-dimensional space. The more the number of samples, the better the classification effect of SVM. This algorithm's time and space complexity are high, and they increase with the number of samples and categories. Further, the algorithm includes many parameters and the choice of parameters significantly affects the classification performance of the model.

Statistics-based models are the mainstream research methods in natural language processing. The most typical classification model is the k-Nearest Neighbor (kNN) model (Bijalwan et al., 2014). Given an unlabeled document "d" the classifier finds the closest "k" adjacent documents of "d". The algorithm determines the category of "d" according to the categories of the "k" documents. This process does not have a specific training and learning process, but only compares the similarity between the unlabeled text and each labeled training set text. The classification performance is often affected by the quality and quantity of the text in the training set. This algorithm's time and space complexity are relatively high, so it is not suitable for tasks with many training samples.

With the development of statistical learning methods and the rapid development of the internet after 1990, online texts have snowballed. Traditional machine learning methods have developed rapidly, gradually forming a process of artificial feature engineering and shallow classification modeling. Traditional machine learning methods divide text classification problems into feature engineering and classifier training. Feature engineering is divided into text preprocessing, feature extraction, and text representation. Its goal is to convert text into a format that can be processed by

a classifier, and add some auxiliary information to help classification. The data processed by feature engineering has better feature expression ability than the original text.

2.2. Artificial Intelligence Methods

The main problem of traditional machine learning methods is that the vector representation of text often has high-dimensional and sparse features, but the expressiveness of features is very weak. The traditional machine method is mainly used for tasks with small data and low cost. After 2010, text classification research gradually changed from the traditional shallow learning mode to the deep learning model. Compared with the shallow learning method, the deep learning method avoids the manual design of rules and features. It obtains the semantic representation of the text through model learning. Unlike shallow models, deep learning methods learn a set of nonlinear transformations that directly map features to outputs, and feature engineering is integrated into the model fitting process. One of the crucial reasons deep learning has achieved great success in images and speech is that the original data of images and speech are continuous, dense, and have local correlations. The most important thing in applying deep learning to solve large-scale text classification problems is to solve text representation and then use network structures like Recurrent Neural Networks (RNN) (Tarwani and Edem, 2017), Convolutional Neural Networks (CNN) (Wang et al., 2020) to automatically obtain feature expression capabilities, remove complicated manual feature engineering, and solve problems end-to-end. Early NLP models relied heavily on feature engineering. With the advent of neural networks for NLP tasks, which combined feature learning with model training, researchers turned their research focus to architecture engineering, that is designing a network architecture that can learn data features.

The text document to be classified is first represented in a numerical form through a process called word embedding (Krusner, 2015), so that it can be further processed by machine learning algorithms. Traditional machine learning algorithms (Khan et al., 2010; Joachims, 2012; Yang et al., 2015; Kowsari et al., 2019) and advanced deep learning algorithms (Minaee et al., 2021) are gaining popularity in text classification. Pre-training of Deep Bidirectional Transformer (BERT) is a state-of-the-art model (Kenton et al., 2019; Koroteev, 2021) pre-trained on large language corpora that has been extensively used in several NLP tasks. Several researchers who found that BERT was significantly undertrained further fine-tuned the model and came up with a novel Robust BERT approach model, named RoBERTa (Liu, 2019). This study focuses on Spanish document classification using the state-of-the-art RoBERTa deep learning model.

3. Datasets

Two labeled datasets are used for training and evaluation of the model. These are briefly described in the following sub-sections.

3.1. News Category Dataset

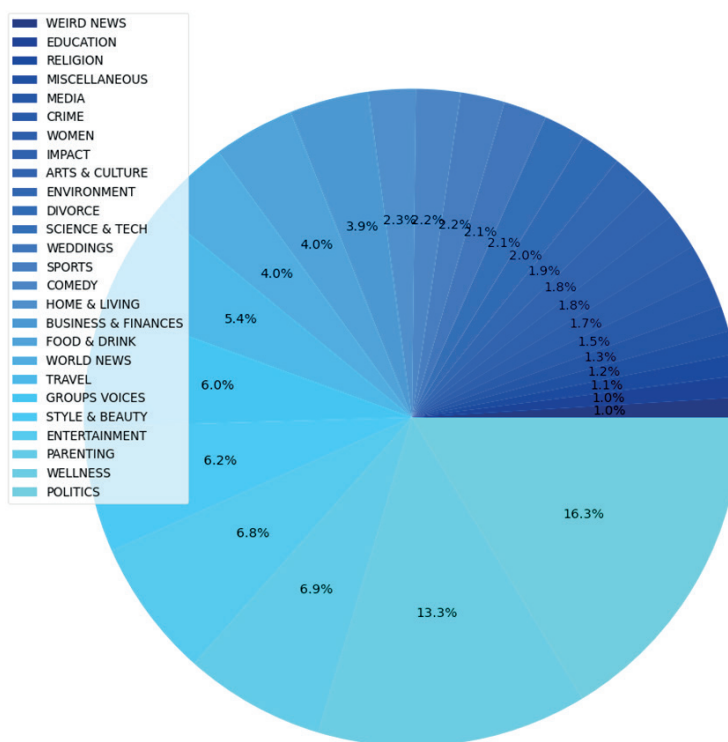
This dataset was on the Kaggle computing competition in 2022. The dataset contains around 210k news headlines from 2012 to 2022 from HuffPost. It is one of the biggest news datasets that can serve as a benchmark for a variety of computational linguistic tasks (Misra et al. 2021; Misra and Rishabh, 2022). Each data record has the attributes tabulated in Table 1.

Table 1
News Category Dataset Attributes

Attribute	Description
Category	Category article belongs to
Headline	Headline of the article
Authors	Persons who authored the article
Link	Link to the post
Short description	Short description of the article
Date	Date the article was published

The dataset has 170,000 news items and contains 41 different News Categories. Figure 1 shows the rather imbalanced distribution of the News Categories. As can be seen from the pie chart, the politic and wellness categories have the most significant proportions, accounting for 16.3% and 13.3%, respectively. However, there are sparsely represented categories such as weird news and education which contain only about 1% of entries.

Figure 1
Class distribution of News Category Dataset



Although the News Category dataset is originally in English, given that it has rich categories and a large amount of data, it is a valuable source of data news classification to train and test relatively large language models. The English contents are translated into Spanish using a separate NLP translation model, and the effect of translation is also evaluated during the training and testing process.

3.2. El mundo

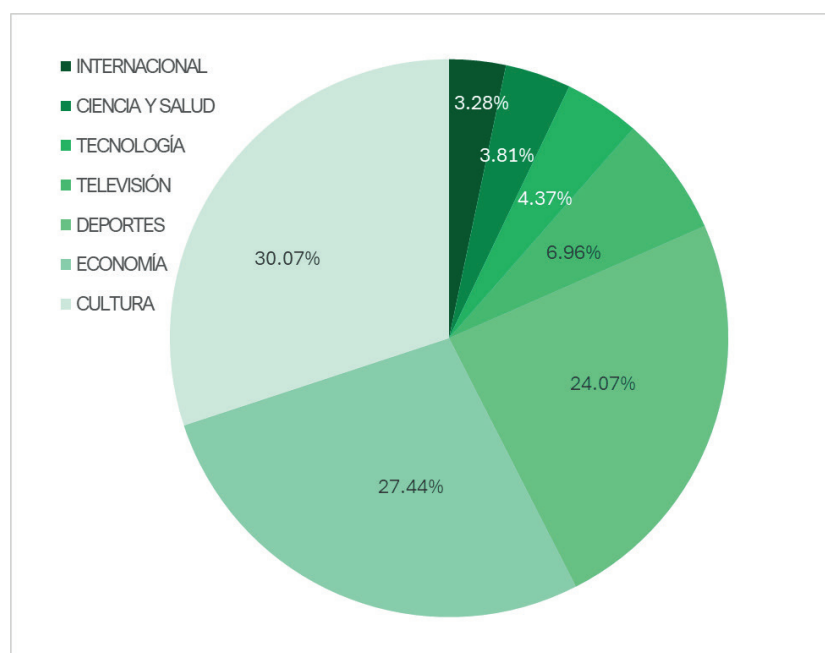
The *El mundo* dataset is accessed from the *El mundo* news site, Spain's second-largest printed daily newspaper. The paper is considered one of the country's newspapers of record along with *El País* and *ABC*. The Scrapy framework is used to crawl news items from September to November 2021. The data records contain the attributes presented in Table 2.

Table 2
El mundo news attributes

Attribute	Description
Category	Category article belongs to
Title	Title of the article
Standfirst	Short description of the index page
Content	The main content of the news

The data set has 10,900 news items and contains seven different News Categories, of which the economic news contains 2,578 records, and the minor international category contains some 323 records. As can be seen from the pie chart (Figure 2), the culture and economic categories have the most significant proportions, accounting for 30.07% and 27.44%, respectively. On the other hand, international and science and health items make up only 3% of the dataset records. Compared to the News Category dataset, the *El mundo* dataset contains only 1/20th of the total number of records. But the contents are all raw Spanish, so this dataset is expected to have a higher quality than the first one. We can also compare the presentation of models between the translated contents and the raw contents during the training process.

Figure 2
Class distribution of El mundo news items



Although there are differences in the total amount of data between the two datasets, they both have the problem of unbalanced data label distribution. Unbalanced datasets are not conducive to model training, as a large number of high-proportion data categories will lead to the fitting direction of the model and the impact of the categories with less data on the model will be weakened. Additional processing in the subsequent data is done to increase the proportion of data for categories with fewer data. This processing helps in the model's training to learn the features of the minority class better.

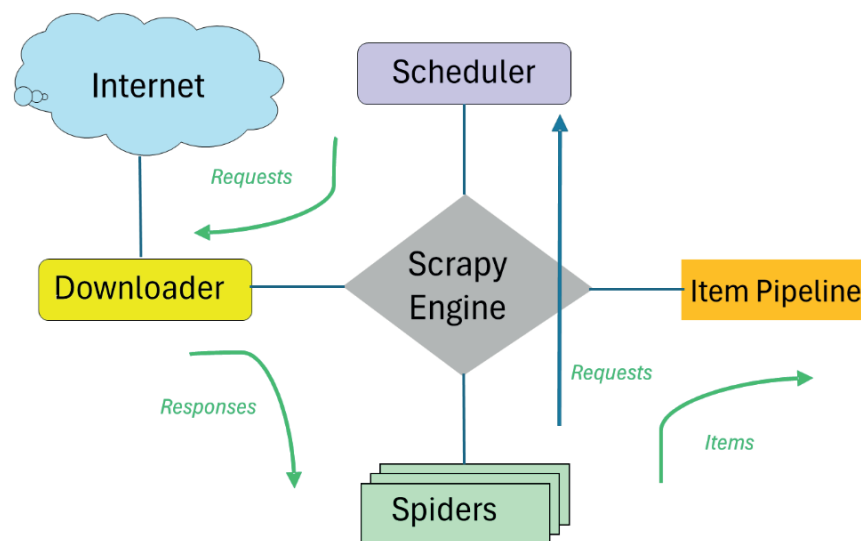
3.3. Data Acquisition

3.3.1. *Crawling and Web-scraping*

Scrapy is a fast and high-level screen scraping and web scraping framework implemented through Python. It can easily and quickly extract structured data from web pages. Users only need to develop specific functional modules to implement crawlers for crawling through the web contents. Scrapy uses the Twisted asynchronous network framework to process network communications, which can speed up data downloading. Compared to the commonly used requests + BeautifulSoup/Selenium in Python, the Scrapy framework has better support for website-level crawlers. This framework also has higher concurrency and has good performance when working with large amounts of data. The functions of each part are as follows: a) engine (it is the core of the entire framework and processes the data flow of the system and triggers transactions); b) scheduler (it accepts the request from the engine, pushes it into the queue, and returns it when the engine requests it again; it can be considered as a priority queue of URLs and at the same time, remove duplicate URLs to avoid repeated crawling); c) downloader (it is used to download web contents and return the web contents to the spider); d) spiders (crawlers are mainly used to extract the information they need from a specific web page, the so-called item; users can also extract links from it and let Scrapy crawl to the next page); e) pipeline (it is responsible for processing the entities extracted by the crawler from the web pages; the main function is to parse the entities, verify the validity of the entities, and remove unnecessary information; when the crawler parses a page, it will be sent to the project pipeline, and the data will be processed in several specific sequences); f) downloader middlewares (the Scrapy engine and the downloader framework mainly process the request and response between the Scrapy engine and the downloader); g) spider middlewares (the main work of the framework between the Scrapy engine and the crawler is to process the response input and request output of the spider); h) scheduler middlewares (the middleware between the Scrapy engine and the dispatcher distributes the request and response sent from the Scrapy engine to the dispatcher).

Figure 3 shows an overview of the Scrapy architecture with its components and an outline of the data flow inside the system (shown by the green arrows). The running process of Scrapy is as follows: (1) the engine fetches a link from the scheduler for the next crawl; (2) the engine encapsulates the URL into a request and sends it to the downloader; (3) the downloader downloads the resource and encapsulates it into a response package; (4) the crawler parses the response, and the parsed item is passed into the pipeline for further processing. If the link to be crawled is parsed, the URL is handed over to the scheduler to wait for the crawling.

Figure 3
Scrapy Structure



Note. Adapted from <http://doc.scrapy.org/en/1.0/topics/architecture.html>

We divided the entire data crawling process into two steps:

First, we crawl the news index. We get the title, category label, and the URL link of the recommended news content by requesting the index page of each category. Further, we get the news query code through the feature code at the end of the link, and then add the code. It is sent to the recommended API in the request parameters. After the API returns the JSON response, we will send the obtained related link as a new request link. This process is recursive, and the filter ensures that we will not process duplicate links. In the end, all currently available news links will be obtained. We pass this data as an index into the pipeline and store it in the *El mundo-index* table.

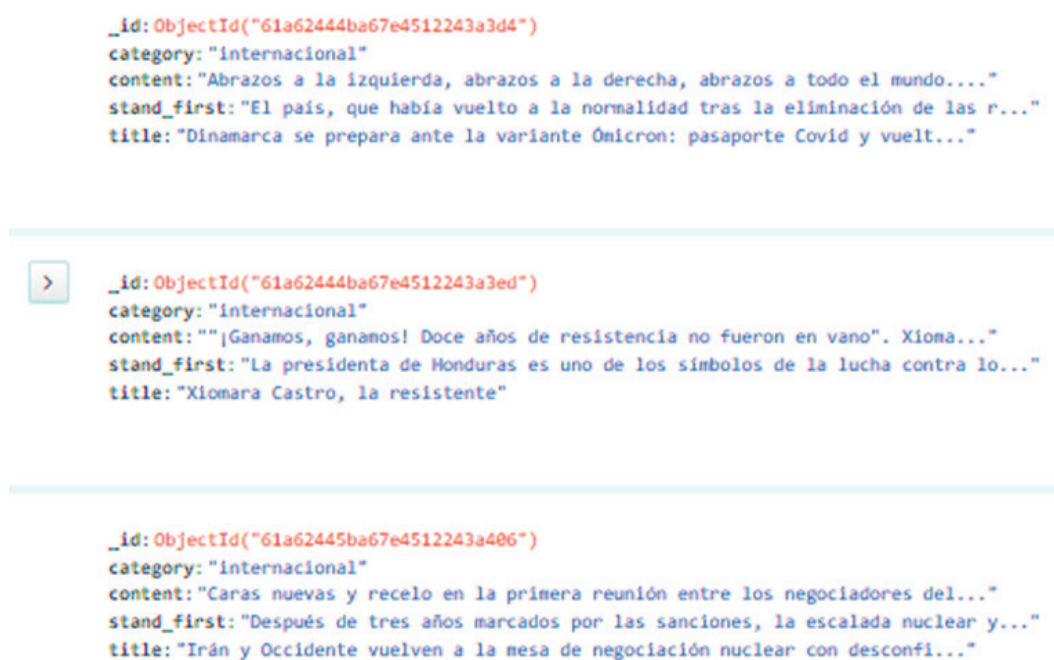
After completing the news index crawling, we obtain the specific news text. By traversing the *El mundo* index database, the link in the index is sent as a request, which will return the HTML response of the particular news page, and we extract what we need through Xpath Title, category, Standfirst, and content. These are passed to the pipeline as detailed news data and stored in the *El mundo* news form. The data in this form will be used as the *El mundo* data set for model training and verification.

3.3.2. Database Storage

The MongoDB database is selected as the storage database for storing the acquired news data. It is a non-relational database and belongs to the document database category. It uses documents as the basic unit of data, and data storage is realized through virtual memory + persistence. MongoDB pays more attention to high-speed read and write speeds than the traditional relational databases that focus on transaction security. For loading large amounts of low-value business data, MongoDB is a better choice. Besides, it has better support for the Python programming language and is suitable for the development of this project.

In our project, we choose MongoDB Atlas, a cloud version for MongoDB. Compared to the local version, the cloud version has the following benefits: (1) easy to maintain as there is no need for local environment configuration; (2) the API port can be quickly accessed through the user authentication key; (3) since the crawling process and training process are executed on different machines, the cloud eliminates the process of data transmission, and the data transmission can also be synchronized in time. Figure 4 shows a section of the database collection.

Figure 4
Database collection



```
_id: ObjectId("61a62444ba67e4512243a3d4")
category: "internacional"
content: "Abrazos a la izquierda, abrazos a la derecha, abrazos a todo el mundo..."
stand_first: "El país, que había vuelto a la normalidad tras la eliminación de las r..."
title: "Dinamarca se prepara ante la variante Ómicron: pasaporte Covid y vuelt..."

>
_id: ObjectId("61a62444ba67e4512243a3ed")
category: "internacional"
content: ""¡Ganamos, ganamos! Doce años de resistencia no fueron en vano". Xiona..."
stand_first: "La presidenta de Honduras es uno de los símbolos de la lucha contra lo..."
title: "Xiomara Castro, la resistente"

_id: ObjectId("61a62445ba67e4512243a406")
category: "internacional"
content: "Caras nuevas y recelo en la primera reunión entre los negociadores del..."
stand_first: "Después de tres años marcados por las sanciones, la escalada nuclear y..."
title: "Irán y Occidente vuelven a la mesa de negociación nuclear con desconfi..."
```

3.4. Data Pre-processing

3.4.1. Data Cleansing

After the data is acquired, we need to pre-process the data before using it for training. Pre-processing mainly includes data cleaning, data conversion, and translation. The translation step was only performed on the News Category dataset. We use an efficient translation model to convert the original English text of this dataset into Spanish.

The data cleaning process mainly involves removing irrelevant symbols and marks in the text. These characters often appear in the obtained network data, including Unicode identifiers, emoji representations, and special symbols. Irrelevant characters are considered OOV (out-of-vocabulary) characters, not presenting semantic information during model training. OOV characters will increase the length of the data text and affect the final training result; so we remove them from the context. For the BERT model, punctuation has a negligible effect on semantics presentation; so we choose to keep only the comma, period, and question mark as the delimiters between short sentences. We use regular expressions to remove emoji and Unicode identifiers. After data cleaning, the length of text in datasets decreased.

Data conversion ensures that the data content meets the requirements of model input. The first step is to sort out the labels. When processing News Category data, we found that the label division is not quite appropriate. For example, there are categories with similar content such as Health living and Wellness; Science and Technology. Our pre-processing step merged these categories into Wellness, and Science & Tech categories. Some categories contain too few samples and can be combined into larger categories, such as Queer voices, Latino voices, Black voices, and Group voices. This process can reduce the training difficulty of model classification and eliminate the imbalance among categories.

The second step is to change the attributes of the data set. We only care about the input text and the corresponding label for classification training. Therefore, other contents in the data set, such as the timestamp and author name are discarded during the pre-processing step. At the same time, the text content may be distributed in different attributes, such as title, stand first, and content. We merged all these into the content attribute.

Finally, we check the length of the data text. We first segment the text based on spaces and convert the string into a word array. We analyzed the length of the word array. The shortest array length is 0 (indicating that there is empty data in the dataset or a crawling failure in the process of crawling). The most extended array length is 1,472. Too short an array contains too little semantic information, which reduces the accuracy of the classification model. We use a filter to remove data with a word array length of less than 10.

Due to the model's input length and GPU memory limitation, we must limit a single input word to 128 words. We applied truncation on data that was beyond this limit. There are three main ways to make truncation: Keeping head, keeping tail, and truncation by sentence. Regardless of the truncation method, the training result of the final model will be different from the original text training result.

3.4.2. Data Augmentation

Traditional machine learning and deep learning methods are widely used in text classification and have achieved significant results. In general, the classification performance of a trained model often depends on the quantity and quality of the training data. However, obtaining a large amount of training data and effectively improving its quality is a daunting task. Real-life datasets being limited in size, are insufficient to produce quality classification results using deep learning methods. Furthermore, as the scale of machine learning models continues to expand and the requirements for task accuracy continue to increase, the size of the required data set is also growing.

Data augmentation is widely studied as an effective solution in this situation. Traditional manual augmentation methods take a lot of time and human resources and, therefore, have become infeasible. This situation has inspired people to research automatic data enhancement methods. Automatic data augmentation has been widely used in computer vision in recent years (Shorten and Khoshgoftaar, 2019). Through the cropping and transformation of images, the amount of data can be rapidly increased. The transformed image can also improve the robustness of the model and ultimately improve the training performance of the model. These methods are particularly effective on small-scale datasets in computer vision (CV) research. Image data augmentation AI techniques

have recently found their way in text data augmentation (Chen, et al., 2021; Wei and Zou, 2019). However, for natural language processing, data enhancement must be done cautiously, owing to the characteristics of natural languages. The change of sentence order and the replacement of keywords can easily cause a change and/or loss of semantic information.

Some studies have used translation models to achieve data augmentation. Translating the original text language into other languages and then translating back to the original language is called back-translation. This process uses multiple or single translation models. Based on the different translation results of the model, the grammatical structure of the original text changes in back-translation. Synonym words also replace some words during this process. Finally, back-translated and original texts are regarded as two different samples for model training. The effect of back-translation largely depends on the quality of the translation model (Koehn, 2005). Poor quality translation can result in the loss of information and semantic changes.

Some studies have proposed an automatic encoder method to generate augmented data. This method trains an automatic encoder to generate text. By inputting candidate text, its sentimental information is processed by the model's attribute discriminator to generate new fake data. The method has been proved to have achieved a certain degree of accuracy improvement on data sets of different scales. However, this method is not widely used in actual research, because the training cost of automatic encoders is very high. To ensure the performance of automatic encoders, a large amount of generalized data needs to be provided for encoder training, which will also take a lot of time and effort. To our knowledge, the evaluation criteria for the augmentation effect of the automatic encoder have not been deeply studied. We adopted the Easy Data Augmentation (EDA), which is a comprehensive augmentation method used in NLP. It obtains augmented data through four text editing methods explained in Table 3 (Wei, 2019). The authors report an average improvement of 0.8% in five NLP tasks performed by training RNN and CNN models.

Table 3
EDA method

Method	Description
Synonym replacement	Randomly extract n words (non-stop words) from the text and replace them with their synonyms.
Random insertion	Select a non-stop word in the sentence and insert a random synonym into a random position in the text. This process is performed n times.
Random swap	Select two words in the text to exchange positions. This process is executed n times.
Random deletion	Random deletion is performed on each word in the text, and the probability of each word being deleted is p .

Compared to the original EDA, we have made the following three improvements in this data augmentation method: (1) The original method is designed for English text and cannot be directly used for Spanish text. The source code uses an English-based stop vocabulary and synonyms library. We modified the source code to use a standard Spanish stop words vocabulary and synonyms library so that the augmentation method could be applied to the Spanish text data. (2) We improved the mechanisms of random insertion and random deletion. Considering that the input text in the data

set may include several short sentences, the word exchange and word insertion between the short sentences are likely to significantly impact the original semantic information. Therefore, we use punctuation marks as separators to limit the exchange of positions between words and the insertion of synonyms within the same short sentence. Such restrictions can reduce the impact of noise insertion on the semantic information of the content. (3) To alleviate the problem of label imbalance, we limited the number of texts generated in different categories. We allowed each category to increase the number of generated contents based on the number of words in the category. This way, we could obtain more content samples on the label with a small number of samples.

Using the EDA method, the size of the News Category dataset increased from 100,000 to 550,000; the size of the *El mundo* dataset increased from 10,900 to 99,000. Using EDA, the prediction accuracy of our trained model has significantly improved as shown in Table 4.

Table 4
EDA result

Model	Train Accuracy	Validation Accuracy
RoBERTa + Raw	72.33	65.44
RoBERTa + EDA	97.45	95.59

4. Deep Learning Spanish Text Classification Results

4.1. Deep Learning Results

We conducted the following detailed experiments to arrive at a comprehensive document classification level.

4.1.1. RoBERTa trained on augmented News Category dataset

The first experiment was extensive data augmentation training as described in the previous section. As the first step of the experiment, we train RoBERTa on the News Category augmented dataset. The News Category dataset has more data than *El mundo*, and the preset number of categories is more. Training tasks face many challenges. The first problem is that the label space and the feature space will be huge, which requires the features to learn enough information density. Secondly, the task of multi-label classification requires that the dataset has a substantial distinction in different categories, because the quality of the data in each category determines the cognitive ability of the model for this category.

We use batch size=128, max category size=30,000 under the setting of 550,000. Five epochs of training were performed on the data set. The accuracy and loss changes and the validation set classification results are shown in Table 5. It can be seen from the results that the accuracy of the small size category has been greatly improved.

4.1.2. RoBERTa trained on translated text

In the second experiment, we used RoBERTa on the News Category data in the English environment set for comparison. The News Category dataset is composed initially of English text. To verify

whether the translation model has harmed the quality of the text, we used the English RoBERTa model to train the untranslated News Category dataset. The model uses the standard Roberta-base model of the Transformers library. We kept the training hyperparameters unchanged. The result on the augmented English dataset is close to the translated one (Table 5). Therefore, we can prove that the translation model has no significant impact on the quality of the text.

4.1.3. RoBERTa trained on *El mundo* dataset

In the third experiment, we trained on the *El mundo* dataset using the RoBERTa model. The *El mundo* data is from Spanish-language news sites, and the text has not been translated; so it should theoretically be of a better quality than News Category. The preset category of this dataset is much smaller than the News Category, and the performance requirements of the model are soft during classification training. We use this training step to validate the model's classification performance on a smaller but high-quality dataset. We first performed training for three epochs on the raw 16,211 data. The fitting degree of the model to the data is shallow, especially for the category with a small sample size (including only a few hundred samples). The accuracy of the category is less than 50%. We believe that this is due to the huge parameter size of the BERT model, and the model parameters cannot be fully trained in the case of too few samples. We next used the augmented dataset for three epochs of training. We set max category size=20,000 and batch size=128 to train on 99,000 samples. The training result is shown in Table 5. The accuracy of the model has been greatly improved.

4.1.4. RoBERTa further pre-training

Further pre-training language models on in-domain data (domain-adaptive pre-training) or task-relevant data (task-adaptive pre-training) before fine-tuning has been shown to improve downstream tasks' performances (Sun et al., 2019; Zhu et al., 2021). We followed the further pre-training step mentioned in the above studies. We used the News Category model obtained in the second step to train *El mundo*. The step of News Category is regarded as further pre-training to help the model train the *El mundo* dataset. We use this step as a validation of the effect of *Further* pre-training.

Further pre-training method is based on the perspective of the domain shift. Since the unlabeled datasets used by the BERT model during pre-training are often texts from multiple domains, the model is believed to have learned a text representation space with a universal presentation in the language during pre-training. However, texts in downstream tasks often do not have universal features. They are often from domain-specific datasets. Therefore, there is a difference in datasets between pre-training and downstream training of the model, which will lead to the problem of domain shift between the target feature space and the feature space obtained by pre-training. In order to eliminate this domain difference, the model needs to migrate its own representation space during fine-tuning to approach the target representation space.

The BERT model has a vast number of parameters, and at the same time, there is a quantity difference between the pre-training dataset and the downstream task dataset, so the domain transfer problem is difficult to eliminate just by fitting the training set. This phenomenon has been proved in many deep neural networks. Further pre-training steps are intended to mitigate this problem. This step imports additional datasets to aid downstream training of the model. The dataset in this step can be performed using unlabeled data or labeled data; so the dataset has a wide range of options.

The model is exposed to more domain-relevant text representation features through this step. This training step can shorten the data volume difference between downstream tasks and pre-training, effectively reducing the performance degradation caused by domain differences.

Table 5
Classification accuracy of RoBERTa training and testing

Training #	Datasets	Training Accuracy (%)	Testing Accuracy (%)
1	News Category (Spanish)	97.45	95.59
2	News Category (English)	97.22	94.84
3	<i>El mundo</i> (Spanish)	98.71	95.93
4	<i>El mundo</i> on News Category (Spanish)	97.40	93.81

4.2. Results Analysis

After data augmentation and hyperparameter optimization, the model achieved high accuracy under the three training methods. The above results show that after fine-tuning the pre-trained model, the model can be used for the classification task of Spanish news texts. However, we found that the results were contrary to our expectations in the *El mundo* and the News Category + *El mundo* datasets. As shown in Table 5, after the further pre-training step, the model's classification accuracy is slightly lower than that obtained by training only on *El mundo*. According to previous studies, the further pre-training step helps the model learn the language representation within the domain. During *El mundo* training, the model can learn the information in the news content faster through the transfer effect in the further pre-training. The following section explains the analysis performed in depth to determine the cause of this anomaly.

4.2.1. CLS Labels

We finally chose to analyze the CLS labels found in the training model. In the design of BERT, the CLS labels are used for text-level natural language tasks such as classification tasks, and the last layer of the corresponding vector is considered to be the semantic representation of the whole sentence (Trabelsi et al., 2021; Yarullin, 2021). Compared to the other positions in the embedding, a CLS label is able to more fairly fuse the semantic information of each word in the text so as to better represent the semantic information of the whole sentence. After 12 layers of the transformer, the CLS label is the weighted average of all the words after attention. In each batch, the CLS label is a matrix of *[batch size, hidden size]*. The final classification result is determined by the CLS label obtained by training. Therefore, the label best reflects the text representation space of the model.

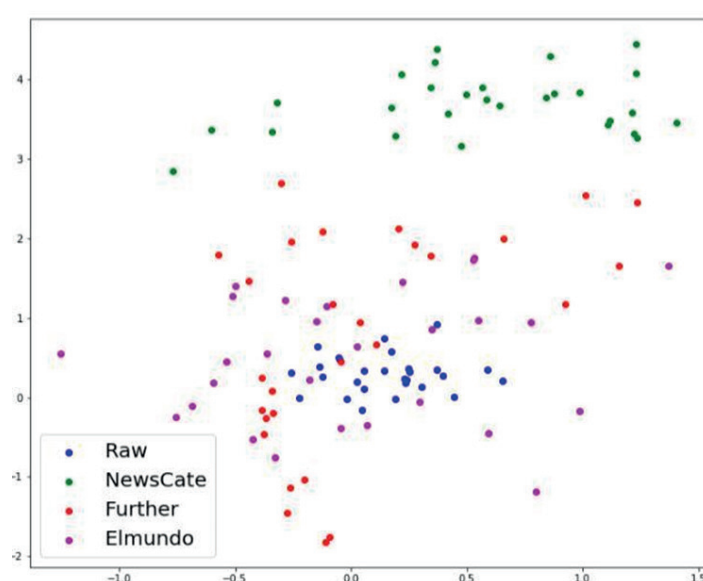
4.2.2. Visualization

Because the dimension of the CLS label is very high, for visualization, we use the MDS algorithm to reduce the dimension of the CLS label (Delicado and Pachón-García, 2024; Huang et al., 2005). The multi-dimensional scaling algorithm determines the representation of high-dimensional objects in the low-dimensional space and makes it as close as possible to match the original similarity. Each point in the high-dimensional space represents an object; so the distance between the points and the similarity between objects is highly correlated. After MDS processing,

two similar objects are represented by two points with similar distances in high-dimensional space, and two different objects are represented by two points with relatively far distances in high-dimensional space. The distance standard of the scale transformation of classical MDS adopts the Euclidean distance.

We select the *El mundo* dataset for evaluation by randomly selecting n samples from each category. We pick a combination of *pretrained model*, *News Category model*, *further pre-trained model*, *El mundo model* for evaluation and loading each model at a time, extracting the CLS label into 2D vector through the MDS function. The result of the sample point distribution is shown in Figure 5.

Figure 5
2D plot of CLS labels

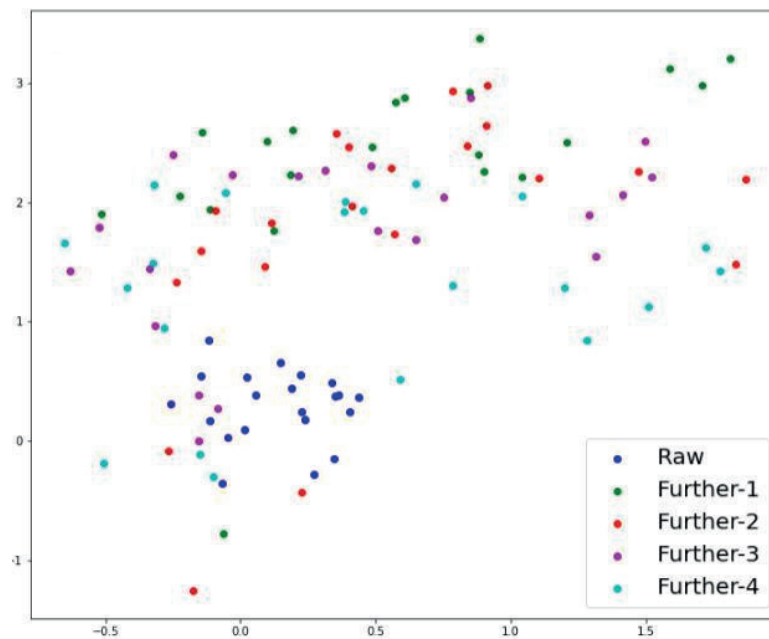


From Figure 5, we can see that after fine-tuning, the CLS label of each model indicates a different degree of migration compared with the original pre-training model. The degree of migration of the News Category model is greater than that of the *El mundo* model. The *El mundo* model representation is closer to the representation of the pre-trained model. The distribution of the further pre-trained model after two training steps has a more neutral representation. The above results show that all models have undergone different degrees of migration. To further explore the changing process of model representation during pre-training, we loaded the parameter records of each model at different epoch stages and repeated the above steps to obtain the CLS label representation of the model at each stage.

Finally, we examine the effect of the further pre-training process (from News Category to *El mundo*) on the CLS labels. It can be observed that as the training progresses, the representation of the model continuously migrates from the cluster of News Category to the cluster of representation of the *El mundo*/raw model. The representation of the final model stops between the two clusters. Note that this intermediate representation is not because the model is still in the under-fitting stage. At this time, the model's training set and validation set accuracy have stabilized and no longer

improved. More training steps did not lead to an increase in the model's accuracy. However, it is foreseeable that as the training continues, the final representation of the model will be further away from the initial News Category representation. The visualization results of the CLS labels are shown in Figure 6.

Figure 6
Further pre-train CLS presentation



The above models have achieved high accuracy on their respective datasets. However, their text representations are quite different. By comparison, we can find that after training on the *El mundo* dataset, the CLS label representation of the model is closer to general presentation. We think this is because the *El mundo* model uses the original Spanish text, which is closer to the Spanish text used in pre-training in terms of grammar and word selection. The News Category model uses translated text. After being translated by the translation model, the initial English texts have received varying degrees of text quality loss. After all, with the current translation model, the translation cannot achieve the quality of the original text. This problem leads to the need for the model to make significant changes to its text representation in downstream tasks. When we used the parameters trained by News Category to train *El mundo*, this text representation was detrimental to the original Spanish text. Adding News Category makes *El mundo* domain migration more difficult when the representation space of News Category is farther away from the pre-training representation space, it is more difficult for *El mundo* to reach the target representation space. This fact makes us unable to improve the model's performance with *Further* pre-training.

5. Conclusion

We demonstrated the development and use of the latest AI tools to achieve high accuracy through text processing, data enhancement, and hyperparameter optimization on the two news datasets. In further analysis, we found that the model made intra-domain migration during training. In the further pre-training step, we found that domain transfer will affect the model's performance.

Improving the selection of further pre-training datasets can alleviate the problem of the intra-domain transfer to a certain extent. We summarize the shortcomings and future prospects of our study as follows:

First, the lack of datasets results in insufficient comparison of experiments. The News Category data set is not the original Spanish dataset. We translated the data set into Spanish data through the translation model for training. However, we found in the subsequent verification that the data in this dataset has the problem of low text quality, which is also an unavoidable problem of the translated text. The data of the *El mundo* dataset performs better in text presentation. However, the problem of insufficient data volume limits our results. We are expected to continue expanding the data set through crawler technology in the follow-up. This step will also effectively alleviate the label imbalance problem in the data set. We used two data sets for comparative experiments in this study, and we believe that such results are insufficient. Adding new datasets for comparative experiments should produce better results if we can add new datasets.

Second, our research on the transfer problem in models lacks quantitative research. We use the MDS algorithm to visually analyze the dimension-reduced vectors of the CLS label vector output by the BERT layer. The MDS algorithm can preserve the space between vectors to a certain extent. However, the application of this algorithm does not consider our prior knowledge of the observed object, lacks the weight processing of the features, we cannot intervene in the processing process in a parameterized way, and the obtained results may not meet the expected results. We assume that each dimension of the vector has the same contribution to the classification task, ignoring the role of MLP, while each dimension may have different effects on MLP. We hope to introduce new quantitative analysis methods in future research to improve migration research credibility further.

References

- Abayomi-Alli, O., Misra, S., Abayomi-Alli, A., and Odusami, M. (2019). A review of soft techniques for SMS spam classification: Methods, approaches and applications. *Engineering Applications of Artificial Intelligence*, 86, 197-212. <https://doi.org/10.1016/j.engappai.2019.08.024>
- Ahmed, H., Traore, I., and Saad, S. (2018). Detecting opinion spams and fake news using text classification. *Security and Privacy*, 1(1), e9. <http://dx.doi.org/10.1002/spy2.9>
- Bijalwan, V., Kumar, V., Kumari, P., and Pascual, J. (2014). KNN based machine learning approach for text and document mining. *International Journal of Database Theory and Application*, 7(1), 61-70. <http://dx.doi.org/10.14257/ijdta.2014.7.1.06>
- Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., and Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, 408, 189-215. <https://doi.org/10.1016/j.neucom.2019.10.118>
- Chen, X., Zhu, D., Lin, D., and Cao, D. (2021). Rumor knowledge embedding based data augmentation for imbalanced rumor detection. *Information Sciences*, 580, 352-370. <https://doi.org/10.1016/j.ins.2021.08.059>
- Delicado, P., and Pachón-García, C. (2024). Multidimensional scaling for big data. *Adv Data Anal Classif*, 18(1), 1-22. <https://doi.org/10.1007/s11634-024-00591-9>
- Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of naacL-HLT*, 1, 4171-4186. <https://doi.org/10.18653/v1/N19-1423>
- Huang, J., Tzeng, G., and Ong, C. (2005). Multidimensional data in multidimensional scaling using the analytic network process. *Pattern Recognition Letters*, 26(6), 755-767. <https://doi.org/10.1016/j.patrec.2004.09.027>
- Jerusha, A., and Rajakumari, R. (2024). Harnessing AI: Enhancing English language teaching through innovative tools. *Proceedings of the 2024 Third International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT)*, 1-7. <https://doi.org/10.1109/ICEEICT61591.2024.10718399>
- Joachims, T. (2012). *Learning to Classify Text Using Support Vector Machines*. Springer Science & Business Media.
- Khan, A., Baharudin, B., Lee, L., and Khan, K. (2010). A review of machine learning algorithms for text-documents classification. *Journal of Advances in Information Technology*, 1(1), 4-20. <http://dx.doi.org/10.4304/jait.1.1.4-20>
- Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. *Proceedings of Machine Translation Aummit X: Papers*, 79-86. <https://aclanthology.org/2005.mtsummit-papers.11>
- Koroteev, M. (2021). BERT: a review of applications in natural language processing and understanding. *ArXiv*. <http://dx.doi.org/10.48550/arXiv.2103.11943>

- Kowsari, K., Jafari, K., Heidarysafa, M., Mendu, S., Barnes, L., and Brown, D. (2019). Text Classification Algorithms: A Survey. *Information*, 10(4), 150. <http://dx.doi.org/10.3390/info10040150>
- Liu, Y. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. <https://doi.org/10.48550/arXiv.1907.11692>
- Medhat, W., Hassan, A., and Korashy, H. (2014). Sentiment Analysis Algorithms and Applications: A survey. *Ain Shams Engineering Journal*, 5(4), 1093-1113. <https://doi.org/10.1016/j.asej.2014.04.011>
- Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., and Gao, J. (2021). Deep Learning Based Text Classification: A Comprehensive Review. *ACM Computing Surveys (CSUR)*, 54(3), 1-40. <https://doi.org/10.1145/3439726>
- Muñoz-Basols, J., Neville, C., Lafford, B., and Godev, C. (2023). Potentialities of Applied Translation for Language Learning in the Era of Artificial Intelligence. *Hispania*, 106(2), 171-94. <https://doi.org/10.1353/hpn.2023.a899427>
- Muñoz-Basols, J., and Fuertes, M. (2024). Opportunities of Artificial Intelligence (AI) in language teaching and learning. In J. Muñoz-Basols, M. Fuertes, and L. Cerezo (Eds.), *Technology-Mediated Language Teaching: From Social Justice to Artificial Intelligence* (pp. 343-360). Routledge.
- Rishabh, M., and Grover, J. (2021). *Sculpting Data for ML: The first act of Machine Learning*.
- Rishabh, M. (2022). *News Category Dataset*. <http://dx.doi.org/10.48550/arXiv.2209.11429>
- Shorten, C., and Khoshgoftaar, T. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1), 1-48. <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0>
- Sun, C., Qiu, X., Xu, Y., and Huang, X. (2019). How to fine-tune bert for text classification? In M. Sun, X. Huang, H. Ji, Z. Liu and Y. Liu (Eds.), *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18-20, 2019 (Lecture Notes in Artificial Intelligence), proceedings 18* (pp. 194-206). Springer. https://doi.org/10.1007/978-3-030-32381-3_16
- Tarwani, K., and Edem, S. (2017). Survey on Recurrent Neural Network in Natural Language Processing. *International Journal of Engineering Trends and Technology (IJETT)*, 48(6), 301-304. <https://doi.org/10.14445/22315381/IJETT-V48P253>
- Xu, K., Liao, S., Li, J., and Song, Y. (2011). Mining comparative opinions from customer reviews for competitive intelligence. *Decision Support Systems*, 50(4), 743-754. <https://doi.org/10.1016/j.dss.2010.08.021>
- Yang, M., Kiang, M., and Shang, W. (2015). Filtering big data from social media—Building an early warning system for adverse drug reactions. *Journal of Biomedical Informatics*, 54, 230-240. <https://doi.org/10.1016/j.jbi.2015.01.011>

Authors contributions

Tad Gonsalves and Hang Hu have participated in the development and data collection. Hang Hu has done software development, result analysis and validation. Tad Gonsalves has done research design, writing, and critical review of the article. Yoshimi Hiroyasu has participated in investigation and data visualization. She has also done the critical review, editing, and revision of the article. All authors approve of the version that is published in the journal.

Acknowledgements

We would like to thank the anonymous reviewers for taking their time to provide valuable feedback.

Funding

This research was funded by the Special Grant for Academic Research Promotion (Priority Area Research), Sophia University, 2020-2022.

Conflict of interest

There are no conflicts of interest.

Correspondence: t-gonsal@sophia.ac.jp

Authors' academic background

Tad Gonsalves is a full professor in the Department of Information and Communication Sciences, Faculty of Science and Technology, Sophia University, Tokyo, Japan. His research areas include bio-inspired optimization techniques and the application of deep learning techniques to diverse problems like autonomous driving, drones, digital art and music, and computational linguistics. Of late, he is also developing Affective Computing models. Gonsalves holds a BS in Theoretical Physics and MS in Astrophysics. He earned his PhD in Information Systems from Sophia University, Tokyo, Japan. His research laboratory (<https://www.gonken.tokyo/>) in Tokyo specializes in applications of deep learning and multi-GPU computing. Gonsalves has published over a hundred and fifty papers in international conferences and journals. He is the author of the book *Artificial Introduction: A Non-Technical Introduction* (2017) Sophia University Press, Tokyo, Japan, and co-author of *Artificial Intelligence for Business Optimization: Research and Applications* (2021), CRC press, London.

Hu Hang obtained the BS in Software Engineering from Beijing University of Posts and Telecommunications, and MS degree in Information Science from Sophia University, Tokyo, Japan. At the under-graduate level, he worked on the development of social network content analysis system with self-designed crawler. He started with crawler based on Scrapy, basic data processing and analysis with various methods, and developed a website with Django for presentation. He also has hands-on experience in developing Mobile and Cloud Applications. His research field is natural language processing, especially the classification of web texts through fine-tuning of pre-trained deep learning models.

Yoshimi Hiroyasu is a full professor at the Center for Language Education and Research (CLER) at Sophia University in Tokyo. She obtained her MA in Linguistics from Sophia University. Since 1989, she has been engaged in the field of teaching Spanish as a Foreign Language (ELE). Her publications include several grammar books, self-study Spanish books, and dictionaries. She has also collaborated on numerous Spanish textbooks, notably *El español y yo* (2013) and *¡Muy bien!* 1 and 2 (2018 and 2019). Currently, she is studying the textual traditions of Spanish teaching in Japan and is developing a corpus of textbooks used in Japan from 1900 to the present.