Using Linguistic Knowledge for Automated Text Identification

Lara Alonso Simón[1,2], José Antonio Gonzalo Gimeno[1], Ana María Fernández-Pampillón Cesteros[1], Marianela Fernández Trinidad[1] and María Victoria Escandell Vidal[1]

[1] *Facultad de Filología, Universidad Complutense de Madrid, c/ Profesor Aranguren, s/n, 28040, Madrid, Spain*
[2] *Corresponding author*

Abstract

This paper describes our proposed classification system to the AuTexTification 2023 Shared Task: Automated Text Identification shared task. The aim of the task is the detection of text automatically generated by six text generation language models or by humans in five domains: legal, how-to articles, tweets, reviews and news. The subtask in which we have participated is a binary classification task with two classes: "human" and "generated". We propose a LinearSVC model where the feeds are represented as tf-idf vectors. Our approach is based on the hypothesis that there are linguistic features according to which, currently, human and generated texts can be characterised. By exploiting those features, automatic classification is efficiently possible. Only four morphological and lexical (character and token n-grams), syntactic (POS tag n-grams) and discourse features (punctuation symbols) have been used for the purpose of this shared task. We describe in this paper the text preprocessing method used, our selection of linguistic features, the different machine learning algorithms with which we have experimented, and the results of the evaluation metrics. Our system achieved the second place, for Spanish, with a macro F1 of 70.6%, while in the same task for English our system achieved the thirteenth place with a macro F1 of 68.33%.

Keywords

Authorship Attribution, Bot Detection, LinearSVC Model, Linguistic Features

1. Introduction

Since the invention of the McCulloch-Pitts neuron [1] and the single-layer perceptron [2], Artificial Intelligence has enormously developed into the cutting-edge neural network models of automatic content generation, such as Generative Pre-trained Transformer (GPT) [3, 4], Pathways Language Model (PaLM) [5] or BigScience Large Open-science Open-access Multilingual Language Model (BLOOM) [6]. These Text Generative Models (TGM) are capable of producing high quality texts: so grammatically correct, fluent and coherent that it is difficult to distinguish them from those written by humans. Text Generative Models have several applications such as conversational response generation, code auto-completion, machine translation or radiology report generation, which have a significant economic and social impact [7].

However, current TGM can also be used to spread fake news, generate fake product reviews, send spam emails for malicious purposes and, in general, they can be used to generate malicious content [8, 9, 10, 7, 11]. These capabilities can have a dangerous social impact, especially when they feed social networks. There are numerous examples of malicious manipulation in the economic context through, for example, fake reviews of products and services [12, 13, 14]. Also in politics and business, bots can generate fake news in order to manipulate and elicit specific responses from society and they can create

IberLEF 2023, September 2023, Jaén, Spain
EMAIL: laraal04@ucm.es (L. Alonso Simón); josgon14@ucm.es (J. A. Gonzalo Gimeno); apampi@filol.ucm.es (A. M.ª Fernández-Pampillón Cesteros); marian37@ucm.es (M. Fernández Trinidad); victoria.escandell@ucm.es (M.ª V. Escandell Vidal)
ORCID: 0000-0002-7278-9122 (L. Alonso Simón); 0000-0002-7565-0644 (J. A. Gonzalo Gimeno); 0000-0002-6606-0159 (A. M.ª Fernández-Pampillón Cesteros); 0000-0002-0087-0829 (M. Fernández Trinidad); 0000-0001-9364-067X (M.ª V. Escandell Vidal)


debates on social media about the behaviour of companies that have negative consequences on their business [15].

It is therefore crucial, at this point in time, to be able to automatically detect between human-written texts and TGM-generated texts [16, 17]. Automatic detection of robot-generated texts, however, is a difficult, and still unsolved, problem that has and will have a great political, economic and social impact [18, 19]. In this respect, evaluation campaigns, such as the shared task in [20], both for English and Spanish, in which this article is embedded, contribute to the research in this area.

In this work we present a solution to the problem of robotic text automatic detection from a linguistic perspective. More concretely, our research questions are:

1. Could explicit linguistic knowledge come in help for automatically detecting generated texts?
2. If so, which linguistic features are discriminant among human-written and automatically generated texts?

Firstly, a review of the state of the art was carried out and a summary of this review is presented in section 2 below. Secondly, the working hypothesis was established, as explained in section 3. Thirdly, a classifier was designed and incrementally implemented based on the hypothesis, as presented in section 4. The results obtained are reasonably satisfactory so that it can be thought that at the moment an efficient way to detect the robotic authorship of the texts can be found on linguistic analysis. Section 5 presents the conclusions of this work and the future directions of research.

The participation in the AuTexTification 2023 shared task [20] (whose aim is precisely the research about automatically detecting generated text) has been an opportunity to begin testing our hypothesis. We are very grateful to the organisers for their idea and their work.

2. Related work

As a result of the social and scientific interest in the detection of generated text, several studies have aimed to the extent to which humans are able to spot texts generated by automatic models. Along with the development and improvements accomplished by the new generation of natural language models, human evaluator's ability to detect generated texts decreases from an accuracy of approximately 70% [21] down to random levels [22, 23], although training can improve detection performance [24]. There is evidence that, even when consisting of well formed texts and outperforming the "Turing test" [25, 26], there might exist some linguistic evidence that could be apprehended by expert linguists [27].

The most commonly used approach to distinguish robot-generated text from human-written text is to formulate the problem as a classification task. The detector (or classifier) created must be accurate, efficient, generalisable, interpretable and robust [7]. One of the current approaches to tackle this task are detectors based on automatic classifiers. [7] present a review of such detectors organised by the underlying methods on which they are based: classifiers trained from scratch, classifiers employing trained TGM that do not require supervised detection examples for further training (zero-shot classifiers), detectors based on a pre trained NLM (Neural Language Model) which is fine-tuned to detect text generated from itself or similar models (these detectors do require supervised detection examples for further training), and human-machine collaborative classifiers.

Previous attempts to address this kind of task, by means of Machine and Deep Learning techniques, can be found in the overview of the CLEF PAN 2019 shared task, an evaluation campaign to encourage research in automatic detection of generated tweets in Spanish and English (as well as in the identification of the gendered of the author) [28]. Within the proposals to solve the assignment, the one that obtained the best accuracy results for both English and Spanish (respectively, 93.6% and 93.33%) was [29], who approached the task by means of a Support Vector Machine, with character n-grams and token n-grams as features.

The corpus of tweets used for that shared task consisted of 3380 and 2400 series of 100 tweets from the same account (for each series), for English and Spanish, respectively. The participants used a wide variety of classification techniques, such as Naïve Bayes, Random Forest, Support Vector Machines or Random Trees algorithms. Among the features which were used to represent the texts were n-grams, stylometry measures (such as lexical density) or different types of embedded vectors. Besides, some systems proposals relied on Deep Learning models, for instance, Convolutional Neural Networks,

Recurrent Neural Networks, Long Short Term Memory Neurons, Feed-Forward Neural Networks or Transformers-based Architectures.

A reading of the overview of CLEF PAN 2019 evaluation campaign [28], taking into account the different proposals and features, as well as the scores obtained, suggests that Machine Learning techniques, and hence, linguistic knowledge, may contribute greatly to the resolution of this kind of task. The Knowledge-based approaches, which are in turn cheaper and more accessible to the majority of laboratories than Deep Learning solutions (in terms of CPU and GPU processing costs), prove themselves essential to the progress of Science and Engineering.

A second approach applied to detect automatically generated text is the use of forensic linguistic techniques to automatically identify robot-generated texts through text analysis. [30] review recent work in forensic analysis for different data modalities, including written texts. They highlight, among others, linguistic features used for the detection of manipulated texts such as the use of punctuation marks, length of headings or grammar. Authorship analysis (attribution or verification) is traditionally based on the extraction of stylometric features of the text, which can be divided into lexical features, character features, syntactic features, structural features and semantic features [30]. [31] points out that, in order to compare written samples and identify the author, lexical, morphosyntactic, punctuation or textual structure variables are observed. She adds that, in addition, complexity variables and the frequency of certain n-grams are taken into account. [32] describes a stylometric analysis to distinguish between tweets written by bots and tweets written by humans. For this case of authorship attribution, seven features were taken into account: type-token ratio, lexical density, hyperlinks, mentions, hashtags, emojis and emojis with a face. Despite their potential effectiveness, defining the set of potentially differentiating linguistic and paralinguistic features is one of the most notable difficulties of this approach.

3. Starting hypothesis

The starting hypothesis is that it is possible to distinguish whether a text has been generated by a TGM or by a person by analysing the values of a set of linguistic and paralinguistic features of the text itself. It is therefore a highly sophisticated version of the "Turing test" [26] to be applied to digital texts. As far as we know, this hypothesis has been tested for the first time in this work for Spanish.

In this paper we have focused specifically on the purely linguistic features of the text. We have chosen, as it will be described in section 4.5, four linguistic features that differentiate automatically generated texts from human-generated texts and on which there is agreement among linguists.

4. Classifier development description

In order to address the resolution of the task proposed by the AuTexTification 2023 organisation, a classifier was developed and evaluated based on the starting hypothesis as well as on the revised related work. The work was done by a team of experts in Semantics and Pragmatics as well as Forensic and Computational Linguistics.

4.1. Working method

For the development of the final solution, a strategy based on three main ideas has been applied: (1) The replica of one of the solutions, previously designed to solve this problem, which achieved good results [29]; the goal is to test to what extent it could be effective in this new dataset. (2) The search for the linguistic features that allow linguists to identify when a text is automatically generated and when it is written by humans; the goal is, according to the working hypothesis, to create a classifier based on these features. And, (3), using the results of (1) and (2), the incremental development of the classifier based on the incremental definition of a vector of linguistic features so that one can accurately assess which linguistic features are really useful for the classifier, i.e., which features really improve the classification accuracy and macro F1 value.

According to this strategy, a working method was designed consisting of the following steps: (1) study and adjustment (preprocessing) of the dataset; (2) selection of linguistic features for the feature vector; (3) construction of the initial classifier with a vector with two features (based on the [29] classifier), and the evaluation of accuracy and macro F1; (4) selection of the classification algorithm based on two features; (5) increase of the classification vector with one more feature, evaluation and decision to keep it or not; (6) iteratively repeating step 5 as long as there are linguistic features to be added and the accuracy and macro F1 results are improved. In more detail, each step is presented below in sections 4.4 to 4.8. Sections 4.2 and 4.3 present the dataset and the setup environment respectively.

4.2. Dataset

The corpus used for the task of automatically detecting generated text was proposed by the AuTexTification 2023 organisation [33]. The complete dataset provided for subtask 1 consisted of a series of 52191 and 55677 texts, respectively for Spanish and English, with texts belonging to five general domaines: legal texts, reviews, WikiHow entries, tweets and news.

For the first phase, the development phase of the solutions, the organisation released 60% of the dataset: 32062 for Spanish and 33845 for English. This initial dataset is composed of texts from only three domains: legal texts, WikiHow entries, and tweets as general domains.

For the second phase, the evaluation phase, the organisation released the other 40% of the dataset: 20129 for Spanish and 21832 for English which are composed of reviews and news, meant to be the evaluation dataset, against which the predictions were submitted to the organisation. Each text within each set of texts included an identification number and the labelled ("human" or "generated") text, except for the evaluation dataset, whose correct labels were publicly available once the campaign results were released.

As it is described in [20], the human texts were crawled from web available sources, such as XLSum, Twitter, EurLex, WikiHow, COAR, COAH and Amazon Reviews. For the generated texts, a prompt extracted from an existing dataset of a given domain and language was used to ask a language model for completion (this completion is just the machine-generated text and the prompt is discarded). The language models were BLOOM (with 1, 3, and 7 billion parameters), and GPT (Babbage, with 1 billion parameters; Curie, with 7 billion parameters; and Text-davinci-003, with 175 billion).

The initial dataset provided to us in phase 1, with 32062 texts for Spanish and 33845 for English, for the classifier development was used as follows: 70% for algorithm training and 30% for testing and metrics evaluation. The corpus splitting was performed randomly in each run of the classification algorithm. The measures of accuracy and macro-F1 value were calculated as the average of the metrics obtained in all runs. To ensure that the training corpus was large enough but, at the same time, the evaluation corpus was sufficiently representative for all seven categories, a split of 70% for training and 30% for evaluation was chosen as the best compromise solution. Other possibilities such as 80-20 for train-test could compromise the need to have all categories in the evaluation corpus or 60-40 could compromise the quality of the classifier by having a training corpus slightly larger than 50%.

4.3. Environment setup

The classifiers were built on Google Colaboratory [34], an online Jupyter notebook environment, with .ipynb for extension of the type of document, so as to allow team members to work collaboratively in an on-line mode. The programming language Python, in its 3.10.11 version, was chosen to code the classifier, because of being of common use among the team members and it has effective libraries for NLP and Machine Learning [35]. Specifically, we used the Python libraries: sklearn [36] (a software machine learning library which interoperate with the Python numerical and scientific libraries NumPy and SciPy), nltk [37] (a suite of libraries for symbolic and statistical natural language processing) and TreeTagger [38] (a tool for annotating text with part-of-speech and lemma information).

4.4. Study and adjustment (preprocessing) of the dataset

The aim of this step is to study what the corpus is like and whether it is suitable for the task to be solved. For this purpose, a textual analysis tool, Sketch Engine [39] which allows semi-automatic analysis of texts, was used. It was observed that (i) texts were short and very short. In the complete dataset, the average range of words is between 52.07 and 62.81 (Table 1). Around 500 texts, for each language, were shorter than 10 words, and there were texts even of only one word length; (ii) many of the texts are incomplete whether at the discourse, sentence or word level; (iii) several texts come from social networks and, in this sense, contain tokens such as hashtags, username mentions, URL links or email addresses. These tokens, if maintained, do not provide statistically significant information because they would appear with a very low frequency. However, their frequency of use could be relevant (do humans use more or less hashtags than robots?).

Table 1
Descriptive statistics of the number of words for the initial and evaluation datasets

Language	Dataset	Number of texts	Word mean	Standard deviation	Min.	Max.	Short (< 10)
Spanish	**Initial**	32062	52.07	27.72	1	131	206
	Evaluation	20129	62.81	19.79	1	136	252
English	**Initial**	33845	53.65	28.66	1	98	239
	Evaluation	21832	62.62	20.79	1	98	356

Therefore, it was decided to preprocess all texts by substituting URL links, username mentions, and hashtags for fixed tags <URLURL>, <UsernameMention> and <HASHTAG>, respectively, following [40]. This step finished at the upholding of the mentioned tags as a lexical feature and the elimination of tokens which would otherwise occur just once.

4.5. Linguistic Features Selection

For the selection of linguistic features from the feature vector, in addition to building a statistical model to detect robotic authorship of a text, it was decided to perform a linguistic analysis of some of the samples (randomly extracted) from the training corpus provided by the AuTexTification organisers. Human-machine collaboration [24, 41, 21] is one of the approaches used to solve the task of determining whether a text has been generated by a robot or written by a human.

Thus, a manual linguistic analysis was carried out in order to obtain the most significant features of the texts that would allow us to classify them into two categories. For this purpose we elaborated a survey, in which 20 randomly sampled texts (10 automatically generated and 10 human) were shown to 5 expert linguists and they were asked to decide if the texts were human-written or machine-generated and to explain the linguistic reasons for such a decision.

Through this manual linguistic analysis the following aspects were observed:

1. Generated texts show a greater repetition of words and sequences than human texts. Humans try to avoid lexical repetition by using synonyms and other strategies such as pronominal substitution.
2. There are more frequent idiomatic expressions in texts written by humans.
3. Robotic texts present the canonical order of SVO constituents almost constantly, while in human-written texts, a higher variation in the order is observed.
4. A larger use of comparative and superlative structures is observed in human texts, probably due to the fact that it is humans who can express prior knowledge and expectations regarding the topic being written about.
5. The discourse markers that appear in generated texts are scarce and repetitive. However, in human texts there is a greater use of discourse markers and connectors.

6. As for punctuation marks, it was observed that in generated texts mostly periods and commas are used, while in human texts there is more variation. In addition, humans use more commas and comparatively fewer periods.

In order to represent these differentiating linguistic features, semantic vectors with word 1-grams (range 2 or 3 for multiword terms) will capture lexical repetition. This feature will be reinforced with character n-gram vectors that will catch roots and morphemes. These two types of vectors will also allow classifiers to detect comparative and superlative structures. Vectors of word n-grams with a range 2-3 will help us to catch idiomatic expressions and the use of discourse markers. The higher or lower variety of punctuation marks, as well as their frequency of use, is captured with the 1-grams of punctuation. Linguistic features at the syntactic level are captured with the n-grams of part-of-speech tags and punctuation symbols. Discourse structure and coherence, characterised by discourse markers and connectors, among others, can be identified with the word n-grams and also with the part-of-speech tags n-grams. POS tags n-grams can also capture idiomatic expressions and comparison structures. Thus, each of the n-gram vectors corresponds to one or more linguistic features.

4.6. Construction of the initial classifier

The aim of this step is the construction of the classifier with a vector with two features (based on the [29] classifier) and the evaluation of macro F1 and accuracy metrics.

An initial compilation of the linguistic features observed in this analysis led us to decide to experiment with linguistic features at the morphological and lexical level. The first prototype of our classification system is based on [29]. The classifier feature vectors were implemented as sparse term frequency-inverse document frequency (tf-idf) of character n-grams and word n-grams. Both tf-idf feature vector representations (for character and token n-grams) were joined using FeatureUnion.

In order to find the best parameters for the feature representations, a hyperparameter tuning was done by hand. The best ranges for the n-grams were (1, 3) for word n-grams and (3, 5) for character n-grams. Some of the results of our trials are shown in Table 2, but it did not improve at all from the metrics with range (1, 3) in word n-grams. Notice that these results are for the algorithms that we subsequently found to give the best results and evaluated with 30% of the training corpus initially provided by the organization of the campaign.

Table 2
Some of the results of the manual hyperoptimization of the initial classifier

Model	Word n-gram range	Macro F1 Spanish	Macro F1 English
Linear SVC	(1, 3)	**83.33**	**83.21**
	(1, 4)	82.88	82.77
	(1, 5)	82.30	82.45
Logistic Regression	(1, 3)	**83.34**	**82.99**
	(1, 4)	82.46	82.72
	(1, 5)	82.46	82.11

4.7. Classifier algorithm selection

In this step of the classifier development the objective is the selection of the classification algorithm using the initial vector of two features. The selection was made by assessing the accuracy and macro F1-value (for Spanish and English) of five traditional Machine Learning algorithms: Linear Support Vector Machine, Logistic Regression, Random Forest, Multinomial Naïve Bayes, and Decision Tree. By using Pipelines, an end to end model was obtained, with the features and the classifier to test. Tables 3 and 4 show the results (the highest values are highlighted in bold). Based on the results, it was decided to select the Linear Support Vector Classification (LinearSVC) and Logistic Regression (LR) algorithms to build the classifier (hence the decision to build two classifiers).

Table 3
Evaluation of the different classification algorithms with tf-idf vectors of character n-grams and token n-grams as features in **Spanish**

Algorithm	Tf-idf n-grams	Macro F1	Accuracy
Linear Support Vector Classification	Character	61.21	67.01
	Word	60.81	66.92
	Character + word	**68.64**	**71.63**
Logistic Regression	Character	63.89	69.9
	Word	61.9	67.2
	Character + word	**69.2**	**71.7**
Random Forest	Character	47.55	60.02
	Word	46.77	59.65
	Character + word	46.55	59.52
Multinomial Naïve Bayes	Character	38.53	54.09
	Word	37.19	56.13
	Character + word	36.45	55.88
Decision Tree	Character	55.55	58.83
	Word	55.15	56.67
	Character + word	55.5	58.25

Table 4
Evaluation of the different classification algorithms with tf-idf vectors of character n-grams and token n-grams as features in **English**

Algorithm	Tf-idf n-grams	Macro F1	Accuracy
Linear Support Vector Classification	Character	66.36	68.41
	Word	62.13	65.49
	Character + word	**67.43**	**69.3**
Logistic Regression	Character	66.14	67.89
	Word	63.45	66.24
	Character + word	**68.07**	**69.54**
Random Forest	Character	42.35	54.84
	Word	40.19	53.87
	Character + word	41.87	54.57
Multinomial Naïve Bayes	Character	38.41	50.87
	Word	36.03	52.1
	Character + word	34.92	51.66
Decision Tree	Character	50.78	55.69
	Word	49.5	53.99
	Character + word	51.13	55.74

4.8. Incremental feature augmentation and evaluation

The following steps were aimed at improving the efficiency of the classifiers by incorporating, one by one, the linguistic features: n-grams of words, n-grams of char, n-grams of POS and punctuation. The computation for each feature was a term frequency-inverse document frequency (tf-idf), it means, the frequency of occurrence of each term in the set of texts [42]. Tables 3 and 4 show the results for Spanish and English respectively. The values shown have been obtained using the evaluation corpus provided by the Autextification organisation in phase 2 of the campaign.

The idea was to test how efficient each feature was on its own and how efficient it was when combined with the others. The results show that combining the traits is more efficient than if they appear

alone. This makes sense linguistically since the combination of the vector features reinforces the identification of the linguistic features. Thus, for example, in the identification of word repetition, the 1-grams of words that have a common root are reinforced by the n-grams of characters that capture the frequency of these common roots.

The augmentation procedure was, basically: (i) first, create the feature vector; (ii) then, the two selected classification algorithms (LinearSVC and LR) were trained; (iii) finally, their accuracy and macro F1 values were computed. To incorporate the POS tags n-gram feature, TreeTagger Python Wrapper [43] was used with the TreeTagger tag repertoire to automatically annotate the texts of the dataset.

As can be seen in Tables 5 and 6 the LinearSVC algorithm provides slightly better results with the full feature vector than Logistic Regression. For this reason, this LinearSVC algorithm has been selected to create the final classifier that has been proposed in the Autextification 2023 campaign. No hyperparameter optimization work was performed and the default ones have been used. Previously we had verified that the default hyperparameters were substantially better than the hyperparameters used by [29]. One of the possible lines of improvement is to carry out a hyperparameter optimization.

Table 5
Evaluation of the Linear Support Vector Classification algorithms with tf-idf vectors of character n-grams, token n-grams, POS tags n-grams, and punctuation symbols as features in **Spanish**

Algorithm	Tf-idf n-grams	Macro F1	Accuracy
Linear Support Vector Classification	Character	61.21	67.01
	Word	60.81	66.92
	POS	60.81	66.92
	Punctuation	47.08	50.76
	Character + word	68.64	71.63
	Character + POS	68.64	71.63
	Character + punctuation	62.11	67.46
	Word + POS	61.53	67.23
	Word + punctuation	65.51	69.54
	POS + punctuation	65.51	69.54
	Character + word + POS	69.21	71.99
	Character + word + punctuation	69.85	72.36
	Character + POS + punctuation	69.85	72.36
	Word + POS + punctuation	66.09	69.85
	Character + word + POS + punctuation	**70.6**	**72.85**
Logistic Regression	Character	63.89	69.9
	Word	61.9	67.2
	POS	61.9	67.2
	Punctuation	47.1	50.88
	Character + word	69.2	71.7
	Character + POS	69.2	71.7
	Character + punctuation	64.71	68.38
	Word + POS	62.16	67.56
	Word + punctuation	64.8	68.93
	POS + punctuation	64.8	68.93
	Character + word + POS	69.06	71.69
	Character + word + punctuation	70.46	72.48
	Character + POS + punctuation	70.46	72.48
	Word + POS + punctuation	64.74	68.99
	Character + word + POS + punctuation	**70.52**	**72.52**

Table 6
Evaluation of the Linear Support Vector Classification algorithms with tf-idf vectors of character n-grams, token n-grams, POS tags n-grams, and punctuation symbols as features in **English**

Algorithm	Tf-idf n-grams	Macro F1	Accuracy
Linear Support Vector Classification	Character	66.36	68.41
	Word	62.13	65.49
	POS	62.13	65.49
	Punctuation	51.16	51.4
	Character + word	67.43	69.3
	Character + POS	67.43	69.3
	Character + punctuation	66.07	68.22
	Word + POS	63.42	66.32
	Word + punctuation	65.23	67.5
	POS + punctuation	65.23	67.5
	Character + word + POS	67.54	69.36
	Character + word + punctuation	67.65	69.4
	Character + POS + punctuation	67.65	69.4
	Word + POS + punctuation	66.26	68.22
	Character + word + POS + punctuation	**68.32**	**69.93**
Logistic Regression	Character	66.14	67.89
	Word	63.45	66.24
	POS	63.45	66.24
	Punctuation	51.16	51.4
	Character + word	68.07	69.54
	Character + POS	68.07	69.54
	Character + punctuation	66.01	67.84
	Word + POS	64.44	66.84
	Word + punctuation	66.05	67.92
	POS + punctuation	66.06	67.92
	Character + word + POS	67.94	69.48
	Character + word + punctuation	67.58	69.21
	Character + POS + punctuation	67.58	69.21
	Word + POS + punctuation	67.01	68.55
	Character + word + POS + punctuation	**67.84**	**69.39**

4.9. Final results and discussion

To evaluate the classifiers, in the first phase of the Autextification task (when the final evaluation corpus was not yet available), the initial corpus was used as follows: 70% for classifier training and 30% for evaluation. The partitioning was done randomly at each run of the program that creates the classifiers. The results, shown in Table 7, have been calculated as the average of fifty runs.

As can be seen in Tables 5 and 6, for each trial (which includes more and more linguistic knowledge) the accuracy and F1 scores improve, confirming that all those levels of linguistic analysis, when in combination, characterise more accurately the generated texts style as opposed to the human-written one.

Table 8 shows the evaluation metrics on the evaluation dataset provided by the organisation in the second phase, which differs from the corpus provided in phase 1 (and with which the classifiers were trained and evaluated) in that it is from different domains, news and reviews, and the text size is slightly larger (Table 1). As expected, values dropped, but the decline was more than ten points, which is a

substantial decline. Despite this, our proposed classification system, Lingüística_UCM classifier, obtained better results than the baselines defined by the task organisers.

Table 7
Macro F1 and accuracy values obtained with a 30% partition of the initial corpus (the remaining 70% was used for training)

AuTexTification Subtask 1	Spanish		English	
	Macro F1	**Accuracy**	**Macro F1**	**Accuracy**
Lingüística_UCM	85.24	85.29	84.97	85

Table 8
Macro F1 values for English and Spanish obtained by the classifier created (Lingüística_UCM) with respect to the base values of the reference classifiers

AuTexTification Subtask 1	Macro F1 Spanish	Macro F1 English
Lingüística_UCM classifier	**70.6**	**68.33**
RoBERTa (BNE) [44]	68.52	-
Logistic Regression	62.4	65.78
Symanto Brain (Few-shot) [45]	56.05	59.44
DeBERTa V3 [46]	-	57.1
Random	50	50
Symanto Brain (Zero-shot) [45]	34.58	43.47

5. Summary, conclusions, and future work

Thanks to a linguistic analysis carried out by expert linguists, it has been possible to extract the most discriminating features when detecting automatically generated texts and thus create a classifier based on traditional machine learning approaches, such as Linear Support Vector Machine and Logistic Regression, which gives good results in the task of automated text identification.

After some minor preprocessing steps of the text dataset, an initial feature selection was carried out based on the linguistic knowledge (morphological, lexical, syntactic and discursive knowledge). The classifier building was based on previous successful systems and the testing of different types of classification algorithms in the search for the one that offered the best results. Once the LinearSVC and LR classification algorithms were selected as the betters, an incremental feature augmentation was carried out and tested. The evaluation of the efficacy of the different feature vectors consisted of the computation of both the arithmetic mean of all the per-class F1 scores and the accuracy.

The results obtained seem to confirm the initial hypothesis: the use of feature vectors based on linguistic features, that differentiate automatically generated texts from manually created texts, allows the creation of automatic classifiers with discriminative capacity at very reasonable costs. Solutions based on traditional statistical Machine Learning classifiers have the advantage over those based on Deep Learning in that they require less computing resources and less energy consumption to build. This facilitates their creation and maintenance at a reasonable economic cost affordable for any organisation or company.

However, the macro F1 values obtained for both Spanish (70.6) and English (68.33) are still low. In this sense, much work remains to be done. We are currently working in Spanish, a more grammatically complex language than English, along two lines:

1. The creation of a comparable corpus of human-robotic in which each text has an average size between 400 and 1500 words so that the discursive features of the texts can be reliably analysed.
2. The textual analysis of the discriminative linguistic features of human-robot authorship of texts for statistical discriminative verification in the corpus.

Once all the discriminative linguistic features have been obtained and tested, as a future work the idea is:

1. to try to explain why the macro F1 value has dropped by more than 10 points when classifying the texts of the evaluation dataset;
2. to rebuild a LinearSVC classifier based on a complete vector of linguistic features and evaluate its effectiveness; and, finally,
3. to explore Deep Learning based solutions that make use of the linguistic features by evaluating their effectiveness with respect to other solutions that do not apply linguistic knowledge.

6. Acknowledgements

This work has been partially funded by the Spanish Ministerio de Educación y Formación Profesional, by means of a collaboration scholarship in the project RobotTalk, for the academic year 2022/2023. This scholarship is granted as part of the Master's Degree in Lingüística y Tecnologías of the Facultad de Filología of the Universidad Complutense de Madrid, in the Departamento de Lingüística y Estudios Árabes, Hebreos y de Asia Oriental, in the Facultad de Filología of the Universidad Complutense de Madrid.

7. References

[1] McCulloch, W., and Pitts, W. (1943). A Logical Calculus of Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, *5*(4), 115-133.
[2] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386-408.
[3] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., and Lowen, R. (2022). Training language models to follow instructions with human feedback. arXiv preprint arXiv:2203.02155.
[4] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.
[5] Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., and Fidel, N. (2022). Palm: Scaling language modeling with pathways. arXiv preprint arXiv:2204.02311.
[6] Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., and Manica, M. (2022). Bloom: A 176b-parameter open-access multilingual language model. arXiv preprint arXiv:2211.05100.
[7] Jawahar, G., Abdul-Mageed, M., and Lakshmanan, L. (2020). Automatic Detection of Machine Generated Text: A Critical Survey. *Proceedings of the 28th International Conference on Computational Linguistics*, 2296-2309.
[8] Bessi, A., and Ferrara, E. (2016). Social bots distort the 2016 US presidential election online discussion. *First Monday*, *21*(11).
[9] Stella, M., Ferrara, E., and De Domenico, M. (2018). Bots increase exposure to negative and inflammatory content in online social systems. *Proceedings of the National Academy of Sciences*, *115*(49), 12435-12440.
[10] Jones, M. O. (2019). The Gulf information war propaganda, fake news, and fake trends: The weaponization of twitter bots in the Gulf crisis. *International Journal of Communication*, *13*(27).
[11] Stiff, H., and Johansson, F. (2022). Detecting computer-generated disinformation. *International Journal of Data Science and Analytics*, *13*, 363-383. https://doi.org/10.1007/s41060-021-00299-5.
[12] Deng, R., and Duzhin, F. (2022). Topological Data Analysis Helps to Improve Accuracy of Deep Learning Models for Fake News Detection Trained on Very Small Training Sets. *Big Data and Cognitive Computing*, *6*(3).
[13] Rodriguez, J., Hay, T., Gros, D., Shamsi, Z., and Srinivasan, R. (2022). Cross-Domain Detection of GPT-2-Generated Technical Text. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1213-1233.
[14] Tourille, J., Sow, B., and Popescu, A. (2022). Automatic Detection of Bot-Generated Tweets. *Proceedings of the 1st International Workshop on Multimedia AI against Disinformation*, 44-51.

[15] Pavlyshenko, B. M. (2022). Methods of Informational Trends Analytics and Fake News Detection on Twitter. https://doi.org/10.48550/arXiv.2204.04891.
[16] Maloyan, N., Nutfullin, B., and Ilyushin, E. (2022). DIALOG-22 RuATD Generated Text Detection. https://doi.org/10.48550/arXiv.2206.08029.
[17] Spanish Government [*Gobierno de España*]. (2021). *Carta de Derechos Digitales*. Plan de Recuperación, Transformación y Resiliencia.
[18] Uchendu, A., Le, T., Shu, K., and Lee, D. (2020). Authorship attribution for neural text generation. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 1808-1822.
[19] Wang, W. Y. (2017). "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2*, 422-426. https://doi.org/10.48550/arXiv.1705.00648.
[20] Sarvazyan, A. M., González, J. Á., Franco Salvador, M., Rangel, F., Chulvi, B., and Rosso, P. (2023). Overview of AuTexTification at IberLEF 2023: Detection and Attribution of Machine-Generated Text in Multiple Domains. In *Procesamiento del Lenguaje Natural*.
[21] Ippolito, D., Duckworth, D., Callison-Burch, C., and Eck, D. (2020). Automatic Detection of Generated Text is Easiest when Humans are Fooled. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 1808-1822.
[22] Clark, E., August, T., Serrano, S., Haduong, N., Gururangan, S., and Smith, N. A. (2021). All That's 'Human' Is Not Gold: Evaluating Human Evaluation of Generated Text. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 7282-7296.
[23] Ethayarajh, K., and Jurafsky, D. (2022). How human is human evaluation? Improving the gold standard for NLG with utility theory. arXiv preprint arXiv:2205.11930.
[24] Dugan, L., Ippolito, D., Kirubarajan, A., and Callison-Burch, C. (2020). RoFT: A Tool for Evaluating Human Detection of Machine-Generated Text. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 189-196.
[25] Bender, E., and Koller, A. (2020). Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5185-5198.
[26] Turing, A. (1950). Computing Machinery and Intelligence. *Mind, 59*(236), 433-460.
[27] Massarelli, L., Petroni, F., Piktus, A., Ott, M., Rocktäschel, T., Plachouras, V., Silvestri, F., and Riedel, S. (2020). How Decoding Strategies Affect the Verifiability of Generated Text. *Findings of the Association for Computational Linguistics: EMNLP 2020*, 223-235.
[28] Rangel, F., and Rosso, P. (2019). Overview of the 7th Author Profiling Task at PAN 2019: Bots and Gender Profiling. In L. Cappellato, N. Ferro, D. E. Losada, and H. Müller (Eds.), *CLEF 2019 Labs and Workshops, Notebook Papers*.
[29] Pizarro, J. (2019). Using n-grams to detect bots on Twitter, notebook for PAN at CLEF 2019. In L. Cappellato, N. Ferro, D. E. Losada, and H. Müller (Eds.), *CLEF 2019 Labs and Workshops, Notebook Papers*.
[30] Bhagtani, K., Yadav, A., Bartusiak, E., Xiang, Z., Shao, R., Baireddy, S., and Delp, E. (2022). An Overview of Recent Work in Media Forensics: Methods and Threats. *2022 IEEE International Conference on Multimedia Information Processing and Retrieval.* https://doi.org/10.48550/arXiv.2204.12067.
[31] Queralt, S. (2020). El uso de recursos tecnológicos en lingüística forense. *Pragmalingüística*, (28), 212-237. http://dx.doi.org/10.25267/Pragmalinguistica.2020.i28.11.
[32] Savoy, J. (2020). Machine learning methods for stylometry: Authorship attribution and author profiling. Springer. https://doi.org/10.1007/978-3-030-53360-1.
[33] Sarvazyan, A. M., González, J. Á., Franco Salvador, M., Rangel, F., Chulvi, B., and Rosso, P. (2023). *AuTexTification Dataset (Full data)* [Data set]. Zenodo. https://doi.org/10.5281/zenodo.7956207.
[34] Google Colaboratory, 2023. URL: https://colab.research.google.com/.
[35] Van Rossum, G., and Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace.
[36] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M.,

Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830. URL: https://scikit-learn.org/stable/.
[37] NLTK Project, 2023. URL: https://www.nltk.org/.
[38] Schmid, H. TreeTagger - a part-of-speech tagger for many languages, 2023. URL: https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/.
[39] Kilgarriff, A., Rychlý, P., Smrž, P., and Tugwell, D. (2004). The Sketch Engine. *Proceedings of the 11th EURALEX International Congress*, 105-116. URL: https://www.sketchengine.eu/.
[40] Daneshvar, S., and Inkpen, D. (2018). Gender Identification in Twitter using N-grams and LSA: Notebook for PAN at CLEF 2018. *CEUR Workshop Proceedings, 2125*.
[41] Gehrmann, S., Strobelt, H., and Rush, A. M. (2019). GLTR: Statistical Detection and Visualization of Generated Text. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 111-116. https://arxiv.org/abs/1906.04043.
[42] Wu, H. C., Luk, R. W. P., Wong, K. F., and Kwok, K. L. (2008). Interpreting TF-IDF term weights as making relevance decisions. *ACM Transactions on Information Systems, 26*(3).
[43] Pointal, L. TreeTagger Python Wrapper's documentation, 2004-2019. URL: https://treetaggerwrapper.readthedocs.io/en/latest/.
[44] Gutiérrez Fandiño, A., Armengol Estapé, J., Pàmies, M., Llop Palao, J., Silveira Ocampo, J., Pio Carrino, C., Armentano Oller, C., Rodríguez Penagos, C., González Agirre, A. and Villegas, M. (2022). MarIA: Spanish Language Models. *Procesamiento del Lenguaje Natural, 68*. URL: https://huggingface.co/PlanTL-GOB-ES/roberta-base-bne.
[45] Symanto, 2023. URL: https://www.symanto.com/nlp-tools/symanto-brain/.
[46] He, P., Liu, X., Gao, J., and Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention. arXiv preprint arXiv:2006.03654. URL: https://huggingface.co/microsoft/deberta-v3-base.