A Convolutional Neural Network Approach for Gender and Language Variety Identification

Helena Gómez-Adorno [a], Roddy Fuentes-Alba [b], Ilia Markov [c], Grigori Sidorov [b], Alexander Gelbukh [b,*]

[a] *Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS), Universidad Nacional Autónoma de México, Mexico*

[b] *Centro de Investigación en Computación (CIC), Instituto Politécnico Nacional, Mexico*

[c] *Institut National de Recherche en Informatique et en Automatique (INRIA), France*

Abstract. We present a method for gender and language variety identification using a convolutional neural network (CNN). We compare the performance of this method with a traditional machine learning algorithm – support vector machines (SVM) trained on character n-grams (n = 3–8) and lexical features (unigrams and bigrams of words), and their combinations. We use a single multi-labeled corpus composed of news articles in different varieties of Spanish developed specifically for these tasks. We present a convolutional neural network trained on word- and sentence-level embeddings architecture that can be successfully applied to gender and language variety identification on a relatively small corpus (less than 10,000 documents). Our experiments show that the deep learning approach outperforms a traditional machine learning approach on both tasks, when named entities are present in the corpus. However, when evaluating the performance of these approaches reducing all named entities to a single symbol "NE" to avoid topic-dependent features, the drop in accuracy is higher for the deep learning approach.

Keywords: Convolutional neural networks, deep learning, author profiling, gender identification, language variety identification, machine learning, character n-grams, Spanish

1. Introduction

Author profiling is the task of identifying certain characteristics of an author, such as age, gender, personality traits, or native language, among others, basing solely on a sample of his or her writings. Gender and language variety identification are considered subtasks of author profiling (AP). The former task aims at identifying the gender of the author (male or female), while the latter is the task of predicting the language variety, in which a given text is written (e.g., Mexican Spanish vs. Peninsular Spanish). Practical applications of these tasks vary from electronic commerce and forensics, where part of the evidence refers to texts (in the case of gender identification (GI)), to machine translation and information retrieval systems (in the case of language variety identification (LVI)).

From the machine learning (ML) perspective, the two tasks can be viewed as multi-class classification problems, when automatic methods have to assign class labels, i.e., author's gender (male/female) or language variety (Mexican Spanish/Peninsular Spanish) to objects (text samples). The most commonly used ML algorithms for solving these tasks are the linear-based classifiers, such as support vector machines (SVM), among others [1]. In traditional ML approaches, character n-gram features have proved to be among the best predictive feature types for both GI [2] and LVI [3,4]. A possible explanation of the effective-

[*] Corresponding author: Alexander Gelbukh. Centro de Investigación en Computación (CIC), Instituto Politécnico Nacional, Mexico. E-mail: gelbukh@gelbukh.com. Web: www.gelbukh.com

ness of these language-independent features consists in their ability to capture lexical and syntactic information, punctuation and capitalization information. Character n-grams can be used either in isolation [3] or combined with other features [5,6], for example, the combination of character n-grams with lexical features (unigrams and bigrams of words) has proved to improve the results for these tasks, including when the Spanish language or its varieties are concerned [5].

In recent years, deep neural networks such as convolutional neural networks (CNN) and recurrent neural networks (RNN) have been widely explored for various natural language processing (NLP) tasks due to their high performance with less need for engineered features [7,8]. Sometimes, more sophisticated architectures are considered as well, for example, hybrid attention networks [9] or deep learning using graph representation [10]. In particular, CNN architectures have shown to be efficient in the AP-related tasks, e.g., personality detection [11] and for authorship attribution [12]. In this paper, we introduce a CNN architecture for gender and language variety identification on a Spanish news corpus and compare it with a traditional ML approach based on the SVM algorithm trained on character and word n-gram features.

Usually different corpora are used to evaluate classification models for GI and LVI, for example, the Spanish language was included in the previous edition of the PAN AP shared task [2] on a corpus composed of Twitter messages, whereas three varieties of Spanish (Argentinian, Peruvian, and Peninsular) were addressed in the recent edition of the VarDial workshop [13] on a corpus of excerpts of journalistic texts. In this work, we use a corpus composed of news articles in Spanish, annotated for the two tasks simultaneously [5]. The corpus covers the following varieties of the Spanish language: Argentinian, Mexican, Colombian, Chilean, Venezuelan, Panamanian, Guatemalan, and Peninsular Spanish.

Following the practice of the VarDial evaluation campaign [14] and other studies [15], we evaluate the extent to which named entities (NEs) affect classifier's performance in these tasks by conducting experiments when reducing all NEs to a single symbol. This allows to evaluate the performance of the examined features avoiding, to some extent, possible topic bias, since NEs are considered to be associated with the thematic area of texts.

The research questions addressed in this work are the following:

(i) Is a CNN able to outperform traditional ML methods in gender and language variety identification?
(ii) Which features and feature combinations are the best predictive for GI and LVI when evaluated on the same corpus in Spanish?
(iii) Which type of embeddings (word or sentence level) contribute to the best CNN model for GI and LVI when evaluated on the same corpus in Spanish?
(iv) What is the impact of NEs on the ML and CNN models performance for these tasks?

The rest of this paper is organized as follows. Section 2 presents the works related to GI and LVI. Section 3 explains the procedure for building the Spanish news corpus and provides its characteristics. Section 4 describes the SVM model, including the feature extraction process and the experimental settings. Section 5 introduces our CNN model. The obtained results and their evaluation are presented in Section 6. Finally, Section 8 draws the conclusions and points to the possible directions for future work.

2. Related Work

Two widely known workshops: PAN[1] and VarDial[2] provide a common platform for researchers interested in evaluating and comparing their systems' performance on the author profiling (AP) and discriminating between similar languages (DSL) tasks (i.e., language variety identification), respectively. The PAN competition is a series of scientific events and shared tasks on digital text forensics, and it is one of the main *fora* regarding the authorship attribution, author profiling, and other authorship analysis-related tasks. The competition has been organized annually since 2009, and it is constantly gaining much attention of researchers from different fields of computational linguistics and natural language processing. The DSL shared task is a part of the VarDial workshop, which is considered the main event regarding language variety identification tasks.

In the 2015 edition of the PAN AP task [16], the winning approach [17] for GI on the Spanish tweets corpus was based on second order attributes technique. In 2016 [2], the shared task focused on cross-gender AP conditions. The best approach [18] in identifying

[1] http://pan.webis.de
[2] http://ttg.uni-saarland.de/vardial2017/sharedtask2017.html

the gender on the Spanish dataset relied on words, sentiment and topic derivation, and stylistic features.

The 2016 edition of the VarDial workshop on DSL [19] used a corpus of short excerpts of news texts, covering Argentine, Castilian, and Mexican Spanish. The overall winner [3] employed character n-gram features (n = 1–7). The last year edition [13] included Argentinian, Peruvian, and Peninsular Spanish. The overall winner of the competition [20] used character n-grams (n = 1–4) for predicting the language group and character n-grams, part-of-speech (POS) n-grams, and proportions of capitalized letters, punctuation marks, and spaces for identifying the language varieties within the group.

In the PAN AP 2017 shared task, there was an increased number of teams that used deep learning techniques to approach the tasks of GI and LVI. Miura et al. [21] presented an architecture composed of a recurrent layer, a convolutional layer and a mechanism for attention. The recurrent layer uses a tweet representation based on word embeddings, while the convolutional layer uses a character embeddings based representation. This system was placed 4^{th} in the official shared task ranking.

A CNN with words bigrams was proposed by Sierra et al. [22]. They obtained an average accuracy of 76% for GI and 95% for LVI on Spanish data. Kodiyan et al. [23] introduced a neural network with a bi-GRU layer followed by an attention mechanism, obtaining GI results of around 78% accuracy for English and Portuguese, and around 71% for Arabic and Spanish.

The results for the DSL task are usually higher than those for AP. For instance, the best performing system [20] in the VarDial 2017 workshop [13] achieved 92.74% of accuracy, while the results for AP under single-genre conditions are usually around 80% [24,25].

3. Corpus

There is a large number of corpora and lexical resources available for the English language, e.g., [26, 27]. However, for Spanish the availability of the corpora is rather scarce, which limits the amount of research for this language. To be able to compare various approaches on the same data, we built a corpus composed of news articles in eight varieties of the Spanish language: Argentinian, Mexican, Colombian, Chilean, Venezuelan, Panamanian, Guatemalan, and Peninsular Spanish.

For the extraction of news articles, we developed a web crawler [28], which is able to automatically navigate through a given website, while discriminating between navigation pages (those pages that contain only links to news) and content pages (those pages that contain news content). After identifying a content page, the crawler extracts the title, author's name, date, and text of the news, and eliminats extra information not related to the news itself (noise), e.g., announcements, related news, navigation menus, etc. The developed crawler requires a set of initial sites (seeds), e.g., *www.eluniversal.com.mx*, *www.reforma.com*, *www.milenio.com*, etc. This set of seeds was manually selected for each of the following countries: Argentina, Mexico, Colombia, Chile, Venezuela, Panama, Guatemala, and Spain.

The crawler navigates through a website in an iterative way extracting the links that belong to valid domains with respect to the given website. At this point, the vast majority of the links not related to the news are filtered. We implemented an extractor of the content of the news, as well as several functions, such as single-linkage clustering, in order to obtain author's name and date of the news using heuristics and regular expressions. Once the crawler has obtained a sufficient number of news for each country, we performed a corpus generation process by evaluating the quality of the extracted news.

The final version of the corpus includes only the news with a minimum size of 750 characters. We removed all the news with distributed authorship, e.g., *AP*, *La prensa*, *Editorial*, etc. Overall, between 10 and 40 texts (news articles) were selected for each author; these ranges were set so that the corpus is not highly unbalanced with respect to the number of documents per author.

We also measured the cosine similarity between all the texts in the corpus in order to remove duplicated entities. For all the cases where the similarity is greater than 0.8, we proceeded to a manual verification of similar news. In the case when there are two or more identical news, we kept only one of them in the corpus. Additionally, we manually checked each news content and deleted names of authors, places, emails, and any other information that may help to reveal the authorship of a text. Finally, during the manual inspection of the corpus, we labeled each text with author's gender (male or female).

The Spanish news corpus is freely available on our website[3]. Table 1 shows the statistics of the corpus by country: number of authors (N of Authors), number of news written by males (Male News) and females (Female News), and the total number of news (Total News).

Table 1
Spanish News Corpus statistics by country.

Country	N of authors	Male news	Female news	Total news
Argentina	21	283	166	449
Venezuela	26	427	401	828
Colombia	25	438	491	929
Guatemala	25	288	310	598
Spain	51	533	375	908
Mexico	35	452	230	682
Panama	29	258	160	418
Chile	20	289	86	375
Total	**232**	**2,968**	**2,219**	**5,187**

The final version of the corpus is composed of multi-labeled 5,187 news articles. In summary, the corpus contains news articles written by 232 different authors categorized according to two genders and eight varieties of Spanish.

4. Support Vector Machine Model

In the following, we briefly explain the features that we used for training the SVM classifier. Then, we focus on the machine learning approach that we used for gender and language variety identification.

4.1. Features

4.1.1. Bag of Words

Bag-of-words (BoW) approach consists in representing the text (document, paragraph, sentence, etc.) by means of the individual units that compose it, that is, the words. The term "words" can refer to:

- The exact text instance.
- The instance of lowercase or uppercase text.
- The word with its part-of-speech tag.
- Word lemma.
- Any other variant of the word.

For example, the text instance (word) "Cars" can be represented as: "Cars", "cars", "cars_N", "car_N", etc.

4.1.2. N-grams

The term n-gram is used to refer to n continuous text units. The term n-grams of words is used to refer to n continuous words in the text.

In the same way when we use the term character n-grams, we refer to the sequence of n continuous characters in the text. The character n-grams can be extracted within the limit of the word without including spaces or from all the text including spaces.

For example, for the sentence "The big red apple":

- The 1-grams (unigrams) of words are the same as the Bag-of-Words: The, big, red, apple.
- The 2-grams (bigrams) of words are: The big, big red, red apple.
- The 3-grams (trigrams) of words are: The big red, big red apple.
- The 2-grams (bigrams) of characters are: Th, he, e_, _b, bi, ig, ..., le.
- The 3-grams (trigrams) of characters are: The, he_, e_b, _bi, ..., ple.

4.2. Experimental Settings

We selected the SVM algorithm, since it is a very popular machine learning method in many NLP-related tasks; moreover, it was the classifier of choice of the majority of the teams in the previous editions of the PAN and VarDial competitions [19,2]. Given that the number of features is much larger than the number of instances, we used the LIBLINEAR [29] library with Crammer and Singer's multi-class support algorithm [30] and default parameters implemented in the WEKA's [31] package.

We evaluated the performance of character and word n-gram features, as well as some of their combinations. Character n-grams vary in order from 3 to 8, while lexical features include unigrams and bigrams of words (without punctuation marks). As feature representation, we used term frequency (TF). It assigns a weight to the term that depends on the number of occurrences of that term in the document or corpus.

We selected a TF threshold greater than or equal to 5, that is, we considered only those features that occur at least 5 times in the corpus. This threshold provided good results for the addressed tasks in several previous studies [24]. Moreover, this threshold value significantly reduces the size of the feature set (on average by approximately 80%).

[3] http://www.cic.ipn.mx/~sidorov/SpanishNewsCorpus.zip

5. Convolutional Neural Network Model

We build an architecture based on the work by Majumder et al. [11], which obtained state-of-the-art results for personality detection. The classification process with the CNN comprises the following intermediate stages: pre-processing, feature learning, and classification. In the following subsections, we explain each of the stages and the architecture of the model, introducing first the concept of word embeddings.

5.1. Word Embeddings

Distributed word representation in a vector space, also known as word embeddings [32], have being studied for more than two decades with Latent Semantic Analysist (LSA) [33] mainly on document level representation. A very popular model architecture for learning distributed word vector representations (Word2Vec) using a neural network was proposed in [34]. This technique captures semantic and syntactic word relations: similar words are close to each other in the vector space. For example, it was shown in [34] that $vector[King] - vector[Man] + vector[Woman]$ results in the vector which is close to the representation of the $vector[Queen]$.

Other two models for building distributed representations are fastText (FTT) [35] and paragraph vectors (PV) [36]. FTT and PV are models inspired by Word2Vec but with certain modifications that give the models specific characteristics and make them promising for different scenarios. FTT adds morphological information to the distributed representations by using characters n-grams. Changing the model in this way makes it more robust when dealing with out-of-vocabulary-words. If a word is not present in the vocabulary of the FTT vectors, one can build a vector for the word by using the different character n-grams vectors learned by the model. PV builds distributed representations for chunks of text, whether they are paragraph, sentences, or whole documents [37].

5.2. Pre-processing

In order to prepare the documents to be fed into the CNN, we carried out several pre-processing steps.

First, we eliminated all the source code of the documents to obtain plain text and lowercased all capital letters. Emoticons, emojis, and hashtags were changed by a special token, as well as urls and numbers. All non-alphanumeric characters were kept and treated as independent tokens. Finally, we built the vectorized representation of each document as follows:

Word-level embeddings representation We used word embeddings to obtain the vectorized representation of a document. All documents were padded to the same length (we used the length of the maximum-length document in the dataset) using a zero vector. Then, the embedding vectors of all the tokens in the document were stacked, obtaining a matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$, where n is the maximum length of the documents in words, and m is the dimensionality of the word vectors.

Sentence-level embeddings representation The documents were treated as lists of sentences and each sentence was considered as a list of words. We used the paragraph vectors (PV) algorithm in order to obtain the vector representation of each sentence in a document. In the same way as described above, documents were padded to the same length at sentence level. Therefore, for each document in the dataset, we have a matrix, $\mathbf{B} \in \mathbb{R}^{p \times m}$, where p is the number of sentences in the document, and m is the dimensionality of the sentence vectors.

We used the same number of dimensions, m, for the word and the sentence representations, since the experiments performed with a different number of dimensions did not yield good results.

5.3. CNN Arquitecture

Our model has eight layers hierarchically stacked where the information flows in the forward direction from the Input layer to the Softmax layer.

1. *Input Layer* This layer expects as input the document vectorized representations built in the preprocessing stage. The only purpose of this layer is to handle the input, checking if it has the expected dimensionality. This layer does not have any trainable parameters.
2. *Concatenation Layer* This layer performs aggregation of the input documents representations by concatenating the word-level and sentence-level representations. With this, for each document we obtain a matrix $\mathbf{S} \in \mathbb{R}^{(n+p) \times m}$.
3. *Convolution Layer* Here the convolution operation takes place. The convolution operation consists of a summation over an element-wise product of a weights matrix called *filter*, $\mathbf{c} \in \mathbb{R}^{d \times m}$,

and each d-gram of the input matrix **S**. The result is a vector called feature map, $\mathbf{r} \in \mathbb{R}^{(n+p)-d+1}$. It is possible to have several filter sizes and multiple filters of the same size in order to capture different features in the input sequences. To the resulting vectors of the convolution we add a bias vector and apply an element-wise non-linear function. We choose the non linearity to be the rectified linear unit (ReLU) function.

4. *1-max Pooling Layer* We perform a 1-max pooling operation in order to capture the most relevant features and to reduce the dimensionality of the feature maps. After this operation we obtain the maximum value per feature map: $\mathbf{o_k} \in \mathbb{R}^{z_k \times 1}$, where k is the filter size, and z is the number of feature maps.
5. *Concatenation Layer* This layer concatenates the features obtained by the previous layer. Thus, obtaining a final vector, $\mathbf{w} \in \mathbb{R}^{(\sum_{k=1}^{n} z_k) \times 1}$ representing the document.
6. *Feedforward Layer* Performs an affine transformation and applies a non-linear function.
7. *Dropout Layer* Applies regularization in order to reduce overfitting.
8. *Softmax Layer* Calculates the probability distribution over the class labels.

5.4. Training and hyperparameter tuning

We trained our CNN model through back propagation with stochastic gradient descent using the Adadelta [38] update rule. We found that the best performing parameters were those shown in Table 2. Word embeddings were kept static in the final evaluation of the model given that fine-tuning them through the CNN training yielded poorer results.

For obtaining word-level embeddings, we evaluated two approaches: word2vec [34] trained on 100 billion words of Google news (for English), and fastTex [35] trained on the Spanish version of Wikipedia (for Spanish). For tokens not found in the pre-trained vectors we used a vector of zeros. For obtaining sentence-level embeddings we used the PV algorithm [36]. We also performed experiments combining word- and sentence-level embeddings.

6. Experimental Results

The evaluation was performed by measuring classification accuracy on the entire corpus under stratified 10-fold cross-validation. Table 3 shows the obtained results for the GI and LVI tasks in terms of classification accuracy (%) under stratified 10-fold cross-validation. For each experiment the number of features (N) is provided. The top accuracy values for each task are shown in bold typeface.

Table 2
CNN model's best performing hyperparameters.

Hyper Parameter	**Values**		
Features maps per filter size	200	300	200
Filters sizes	1	2	3
Number of epochs	60		
Dropout probability	0.6		
Batch size	50		
Number of units in MLP hidden layer	80		
PV number of training epochs	3		
PV dimensionality	300		

As one can see from Table 3, in the experiments with the SVM classifier, higher-order character n-grams (n = 5–8) outperform both lower-order character n-grams and unigrams and bigrams of words for both tasks when evaluated in isolation. The combination of all word and character n-grams provides the best results for the LVI task when using the ML approach, i.e., the SVM classifier.

Moreover, it can be seen that the results obtained with the SVM classifier continue to improve when adding higher-order character n-grams to the combination of features. However, higher-order character n-gram features significantly increase the size of the feature set, especially when used in combinations with each other, and consequently, the computational cost of the training process, while the accuracy improvement is only marginal. Therefore, we limited our SVM experiments with the maximum order of 8 for character n-grams.

It can be also noted that the best model for the GI and LVI tasks only slightly outperforms the bag-of-words approach (1.62% and 1.24%, respectively). This indicates that word unigrams, when used in isolation, are already a challenging baseline for these tasks. In our experiments, word unigrams even outperform character 3-gram features, which are considered to be highly predictive for many NLP tasks [24].

With respect to the CNN models, the experiments showed that the combination of word- and sentence-level embeddings (W2V-PV-CNN and FTT-PV-CNN) achieved higher results than using only one or another in isolation. Although it is noteworthy that adding the PV embeddings to the model that only uses word2vec

Table 3

Accuracy results (%) for the ML and the CNN models in the GI, and LVI tasks.

Word unigrams	**Word bigrams**	**Char. 3-grams**	**Char. 4-grams**	**Char. 5-grams**	**Char. 6-grams**	**Char. 7-grams**	**Char. 8-grams**	**Accuracy (%)**		
Features								**GI**	**LVI**	**N**
✓								73.99	92.92	38,360
	✓							73.13	91.05	94,501
		✓						69.87	91.50	25,631
			✓					72.80	93.75	83,917
				✓				73.92	93.75	189,240
					✓			74.94	94.04	336,422
						✓		75.11	94.04	498,014
							✓	75.61	93.64	628,180
Combinations								**GI**	**LVI**	**N**
✓	✓							74.78	92.94	132,861
✓	✓	✓						71.68	92.60	158,492
✓	✓	✓	✓					72.97	93.14	242,409
✓	✓	✓	✓	✓				73.51	93.45	431,649
✓	✓	✓	✓	✓	✓			74.15	93.70	768,071
✓	✓	✓	✓	✓	✓	✓		74.57	93.99	1,266,085
✓	✓	✓	✓	✓	✓	✓	✓	75.28	94.16	1,894,265
CNN Model								**GI**	**LVI**	**N**
W2V-CNN								73.64	93.33	800
FTT-CNN								75.47	93.98	800
W2V-PV-CNN								73.30	92.99	800
FTT-PV-CNN								**75.90**	**94.50**	800

vectors (W2V-CNN) did not make a significant improvement. Nevertheless, FTT-PV-CNN (fastText and paragraph vectors) model is the best performing model on GI and LVI tasks with classification accuracy of 75.9% and 94.5%, respectively.

As expected (see Section 2), the results for the LVI task are significantly higher than those for GI with both approaches, SVM and CNN.

Following the practice of the VarDial workshop [14], we conducted additional experiments reducing all named entities (NEs) to a single symbol (*#NE#*) to evaluate their impact on these tasks. In order to extract NEs, we used a Spanish model of the Stanford named entity recognizer (NER) [39]. The version of the corpus with reduced NEs is also available on our website[4]. The results for both SVM and CNN approaches after reducing NEs to a single symbol, as well as the accuracy drop and the best result (in bold typeface) for each experiment are provided in Table 4.

As one can see comparing Tables 3 and 4, features' performance using the SVM classifier shows similar behavior when NEs are reduced, that is, higher-order character n-grams outperform both lower-order character n-grams and lexical features. Moreover, higher-order character n-grams seem better able to cope with the setting when topic-dependent information is discarded than lexical features, that is, the average drop in accuracy for character n-grams is 2.52% when for the lexical features is approximately 5%.

The average accuracy drop after reducing NEs 2.47% for GI and 3.72% for LVI. For LVI the accuracy drop of 3.72% is higher than the one of around 2% reported in the VarDial workshop proceedings [14]. One of the possible explanations is the nature of our corpus, which contains much longer texts than the VarDail corpus of excerpts of journalistic texts.

[4] http://www.cic.ipn.mx/~sidorov/SpanishNewsCorpus.zip

Table 4

Accuracy results (%) for the ML and the CNN models in the GI, and LVI tasks after reducing all NEs to a single symbol. Accuracy drop (%) with respect to the results in Table 3 is provided for each experiment.

Word unigrams	Word bigrams	Char. 3-grams	Char. 4-grams	Char. 5-grams	Char. 6-grams	Char. 7-grams	Char. 8-grams	Accuracy (%)			Accuracy Drop (%)	
Features								**GI**	**LVI**	**N**	**GI**	**LVI**
✓								70.79	86.70	31,884	3.20	6.22
	✓							70.43	84.36	89,448	2.70	6.69
		✓						66.96	86.97	19,209	2.91	4.53
			✓					70.10	90,57	62,514	2.70	3.18
				✓				71.43	91.34	148,026	2.49	2.41
					✓			72.45	91.65	285,365	2.49	2.39
						✓		73.55	91.77	445,546	1.56	2.27
							✓	**73.90**	91.61	579,349	1.71	2.03
Average Accuracy Drop (%):											2.47	3.72
CNN Model								**GI**	**LVI**	**N**	**GI**	**LVI**
FTT-PV-CNN								69.90	**92.07**	800	6.00	2.43
FTT-CNN								68.76	84.85	800	6.71	9.13

Even though, the classification accuracy of the CNN models still achieve the highest results for the LVI task, the accuracy drop of the GI task when NEs are reduced are three times larger than for the SVM models. This behavior yielded a low performance of the CNN model for this task when NEs are reduced.

7. Experiments on another corpus (English data)

In order to evaluate the robustness of our CNN approach on another corpus and language (English), we performed experiments on the PAN AP 2017 shared task training dataset [1]. The task consisted in predicting gender and language variety in Twitter. The training corpus covers the following languages: English, Spanish, Portuguese, and Arabic.

We conducted experiments on the provided PAN AP 2017 training dataset under 10-fold cross-validation with the CNN approach for only the English and Spanish subsets. The varieties included in the English corpus are: Australia, Canada, Great Britain, Ireland, New Zealand, and United States. The Spanish corpus included the following varieties: Argentina, Chile, Colombia, Mexico, Peru, Spain, Venezuela.

We compare the results of our CNN approach with one of the top systems in the competition [40]. This was the only work that provided cross-validation results in the working-note paper. We compare the results on the training corpus under 10-fold cross-validation, since the test set was not made available by the competition organizers. The system presented in [40] uses a ML approach with different types of features, such as word and character n-grams, and parameter configurations depending on the language. The 10-fold cross-validation (10FCV) results in terms of classification accuracy on the PAN AP 2017 training corpus are shown in Table 5.

Table 5

CNN model results (accuracy, %) on the PAN AP 2017 data.

Model	English		Spanish	
	GI	LVI	GI	LVI
Markov et al. [40]	82.11	87.19	80.00	95.31
W2V-CNN	**83.88**	87.66	**82.33**	94.76
FTT-CNN	83.55	**89.44**	81.33	**95.47**
W2V-PV-CNN	80.39	78.92	77.52	78.92
FTT-PV-CNN	80.44	78.50	77.55	91.62

It can be observed that the CNN approach obtains higher accuracy than the ML approach for both GI and LVI in all cases. The average accuracy improvement of the best CNN approach over the ML approach presented in [40] is higher on the GI task than on the LVI task (2.0% and 1.2%, respectively). With respect to the embeddings features, we found that the performance of the W2V-CNN model outperformed the other models for GI, while the FTT-CNN model achieved the best results for the LVI on both languages.

8. Conclusions and Future Work

In this paper, we examined the performance of a ML model trained on traditional features and a CNN model trained on word- and sentence-level embeddings on the tasks of gender and language variety identification. For the evaluation of the proposed models we used a multi-labeled corpus of news articles in different varieties of Spanish. Each news article in the corpus is annotated with the author's name, author's gender, and one of 8 varieties of the Spanish language.

The obtained results indicate that when using the ML approach, higher-order character n-grams outperform lower-order character n-grams for the two tasks and provide the best results for gender identification when used in isolation (75.61% of accuracy). The combination of all word and character n-grams of different orders (n = 1–2 for words and n = 3–8 for characters) outperforms other combinations of such features and provides the best results for LVI (94.16%).

On the other hand, the results obtained with the deep learning approach show that the CNN model trained on fastText (FTT) embeddings and the combination of FTT embeddings and Paragraph Vector (PV) embeddings outperformed the CNN model trained on word2vec embeddings and the combination of word2vec and PV embeddings for the Spanish news corpus.

When comparing the performance of the CNN model with the ML model, we note that the CNN slightly outperforms the ML by 0.29% for gender identification and by 0.34% for language variety identification. The obtained results are promising, considering the relatively small size of the corpus.

We also evaluated the impact of named entities on these tasks. Our results showed that reducing them all to a single symbol "NE" to avoid topic-dependent features decreases accuracy by around 2.5%–3.7% , depending on the task. We also observed that the traditional ML approach is more robust than the CNN model when NEs are reduced.

Additional experiments on the multi-language corpus for author profiling (PAN 2017) showed that word-level embeddings (word2vec or fastText) outperform a traditional ML approach, which was ranked among the top approaches in the PAN author profiling shared task 2017. However, the combination of word- and sentence-level embeddings decreased the performance for both classification tasks.

In general, the CNN models outperformed traditional ML models by a low margin in the majority of our experiments. On the Spanish News corpus the CNN model obtained an average of 0.31% of improvement over the ML model, while on the PAN AP 2017 corpus the average improvement was 1.62%. Taking into consideration the small size of the evaluation corpus, this results are encouraging and it is worth continuing with this line of research.

One of the directions for future work would be to examine the performance of other types of embeddings, including embeddings learned on word and character n-grams of various sizes [37]. Moreover, we will examine the contribution of different pre-processing steps, as well as conduct experiments using other neural network algorithms, such as recurrent neural networks. Finally, the performance of the CNN evaluated in this work will be tested on other corpora, including a cross-genre scenario.

Acknowledgements

The work was done with partial support of the Mexican Government via the CONACYT project 240844 and Instituto Politécnico Nacional grants SIP-20181849, SIP-20171813, and SIP-20181792. The work was done when A. Gelbukh was visiting the Research Institute for Information and Language Processing, University of Wolverhampton, on a grant from the Sabbatical Year Program of the CONACYT, Mexico.

References

[1] F. Rangel, P. Rosso, M. Potthast and B. Stein, Overview of the 5th Author Profiling Task at PAN 2017: Gender and Language Variety Identification in Twitter, in: *Working Notes Papers of the CLEF 2017 Evaluation Labs*, CEUR-WS.org, 2017.

[2] F. Rangel, P. Rosso, B. Verhoeven, W. Daelemans, M. Potthast and B. Stein, Overview of the 4th Author Profiling Task at PAN 2016: Cross-genre Evaluations, in: *Working Notes Papers of the CLEF 2016 Evaluation Labs*, CEUR-WS.org, 2016, pp. 1–26.

[3] C. Çöltekin and T. Rama, Discriminating similar languages: experiments with linear SVMs and neural networks, in: *Proceedings of the 3rd Workshop on NLP for Similar Languages, Varieties and Dialects*, VarDial'16, 2016, pp. 15–24.

[4] H. Gómez-Adorno, I. Markov, J. Baptista, G. Sidorov and D. Pinto, Discriminating between Similar Languages Using a Combination of Typed and Untyped Character N-grams and Words, in: *Proceedings of the 4th Workshop on NLP for Similar Languages, Varieties and Dialects*, VarDial'17, 2017, pp. 137–145.

[5] M.A. Sanchez-Perez, I. Markov, H. Gómez-Adorno and G. Sidorov, Comparison of Character n-grams and Lexical Features on Author, Gender, and Language Variety Identification on the Same Spanish News Corpus, in: *International Conference of the Cross-Language Evaluation Forum for European Languages*, CLEF'17, Springer, 2017, pp. 145–151.

[6] I. Markov, H. Gómez-Adorno, G. Sidorov and A. Gelbukh, The Winning Approach to Cross-Genre Gender Identification in Russian at Rusprofiling 2017, in: *FIRE 2017 Working Notes*, FIRE'17, 2017, pp. 20–24.

[7] Y. Kim, Convolutional Neural Networks for Sentence Classification, in: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, EMNLP'14, 2014, pp. 1746–1741.

[8] P. Blunsom, E. Grefenstette and N. Kalchbrenner, A convolutional neural network for modelling sentences, in: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, ACL, 2014, pp. 655–665.

[9] Y. Zhou, J. Xu, J. Cao, B. Xu, C. Li and B. Xu, Hybrid Attention Networks for Chinese Short Text Classification, *Computación y Sistemas* **21**(4) (2017), 759–769.

[10] M.G. Sohrab, T. Nakata, M. Miwa and Y. Sasaki, EDGE2VEC: Edge Representations for Large-Scale Scalable Hierarchical Learning, *Computación y Sistemas* **21**(4) (2017).

[11] N. Majumder, S. Poria, A. Gelbukh and E. Cambria, Deep Learning-Based Document Modeling for Personality Detection from Text, *IEEE Intelligent Systems* **32**(2) (2017), 74–79.

[12] S. Ruder, P. Ghaffari and J.G. Breslin, Character-level and Multi-channel Convolutional Neural Networks for Large-scale Authorship Attribution, *arXiv preprint arXiv:1609.06686* (2016).

[13] M. Zampieri, S. Malmasi, N. Ljubešić, P. Nakov, A. Ali, J. Tiedemann, Y. Scherrer and N. Aepli, Findings of the VarDial Evaluation Campaign 2017, in: *Proceedings of the 4th Workshop on NLP for Similar Languages, Varieties and Dialects*, VarDial'17, 2017, pp. 1–15.

[14] M. Zampieri, L. Tan, N. Ljubešić, J. Tiedemann and P. Nakov, Overview of the DSL shared task 2015, in: *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects*, LT4VarDial'15, 2015, pp. 1–9.

[15] G. Ríos, G. Sidorov, N. Castro, A. Nava and L. Chanona-Hernández, Relevance of Named Entities in Authorship Attribution, in: *Proceedings of the 15th Mexican International Conference on Artificial Intelligence (MICAI'16)*, LNAI, Springer, 2017.

[16] F. Rangel, F. Celli, P. Rosso, M. Pottast, B. Stein and W. Daelemans, Overview of the 3rd Author Profiling Task at PAN 2015, in: *CLEF 2015 Labs and Workshops, Notebook Papers*, Vol. 1391, CEUR.org, 2015, pp. 1–18.

[17] M.A. Álvarez-Carmona, A.P. López-Monroy, M. Montes-y-Gómez, L. Villaseï£¡or-Pineda and H. Jair-Escalante, INAOE's Participation at PAN'15: Author Profiling task, in: *Working Notes Papers of the CLEF 2015 Evaluation Labs*, Vol. 1391, CEUR.org, 2015.

[18] P. Gencheva, M. Boyanov, E. Deneva, P. Nakov, G. Georgiev, Y. Kiprov and I. Koychev, PANcakes Team: A Composite System of Genre-Agnostic Features For Author Profiling, in: *Working Notes Papers of the CLEF 2016 Evaluation Labs*, CEUR-WS.org, 2016, p. 7.

[19] S. Malmasi, M. Zampieri, N. Ljubešić, P. Nakov, A. Ali and J. Tiedemann, Discriminating between Similar Languages and Arabic Dialect Identification: A Report on the 3rd DSL shared task, in: *Proceedings of the 3rd Workshop on Language Technology for Closely Related Languages, Varieties and Dialects*, VarDial'16, 2016, pp. 1–14.

[20] Y. Bestgen, Improving the Character N-gram Model for the DSL Task with BM25 Weighting and Less Frequently Used Feature Sets, in: *Proceedings of the 4th Workshop on NLP for Similar Languages, Varieties and Dialects*, VarDial'17, 2017, pp. 115–123.

[21] Y. Miura, T. Taniguchi, M. Taniguchi and T. Ohkuma, Author Profiling with Word+Character Neural Attention Network, in: *CLEF 2017 Evaluation Labs and Workshop – Working Notes Papers*, CEUR-WS.org, 2017.

[22] S. Sierra, M. Montes-Y-Gómez, T. Solorio and F. González, Convolutional Neural Networks for Author Profiling in PAN 2017, in: *CLEF 2017 Evaluation Labs and Workshop – Working Notes Papers*, CEUR-WS.org, 2017.

[23] D. Kodiyan, F. Hardegger, S. Neuhaus and M. Cieliebak, Author Profiling with Bidirectional RNNs using Attention with GRUs, in: *CLEF 2017 Evaluation Labs and Workshop – Working Notes Papers*, CEUR-WS.org, 2017.

[24] U. Sapkota, S. Bethard, M. Montes-y-Gómez and T. Solorio, Not All Character N-grams Are Created Equal: A Study in Authorship Attribution, in: *Proceedings of the 2015 Annual Conference of the North American Chapter of the ACL: Human Language Technologies*, NAACL-HLT'15, 2015, pp. 93–102.

[25] I. Markov, H. Gómez-Adorno, J.- Posadas-Durán, G. Sidorov and A. Gelbukh, Author Profiling with doc2vec Neural Network-based Document Embeddings, in: *Proceedings of the 15th Mexican International Conference on Artificial Intelligence*, MICAI 2016, Vol. 10062, LNAI, Springer, 2017, pp. 117–131.

[26] E. Stamatatos, On the robustness of authorship attribution based on character n-gram features, *Journal of Law & Policy* **21**(2) (2013), 427–439.

[27] G. Sidorov, M. Ibarra Romero, I. Markov, R. Guzman-Cabrera, L. Chanona-Hernández and F. Velásquez, Detección automática de Similitud entre Programas del Lenguaje de Programación Karel basada en técnicas de Procesamiento de Lenguaje Natural, *Computación y Sistemas* **20**(2) (2016), 279–288.

[28] F.V. Jiménez, M.A. Sánchez-Pérez, H. Gómez-Adorno, J.P. Posadas-Durán, G. Sidorov and A.F. Gelbukh, Improving the Boilerpipe Algorithm for Boilerplate Removal in News Articles Using HTML Tree Structure, *Computación y Sistemas* **22**(2) (2018).

[29] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang and C.-J. Lin, LIBLINEAR: A Library for Large Linear Classification, *Journal of Machine Learning Research* **9** (2008), 1871–1874.

[30] K. Crammer and Y. Singer, On the Learnability and Design of Output Codes for Multiclass Problems, *Machine Learning* **47**(2–3) (2002), 201–233.

[31] I. Witten, E. Frank, M. Hall and C. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*, 4th edn, Morgan

Kaufmann, 2016.

[32] J. Turian, L. Ratinov and Y. Bengio, Word representations: A simple and general method for semisupervised learning, in: *Proceedings of the 48thAnnual Meeting of the Association for Computational Linguistics*, 2010, pp. 384–394.

[33] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer and R. Harshman, Indexing by latent semantic analysis, *Journal of the American society for information science* **41**(6) (1990), 391.

[34] T. Mikolov, K. Chen, G. Corrado and J. Dean, Efficient Estimation of Word Representations in Vector Space, *arXiv preprint arXiv:1301.3781* (2013), 1–12.

[35] P. Bojanowski, E. Grave, A. Joulin and T. Mikolov, Enriching Word Vectors with Subword Information, *Transactions of the Association for Computational Linguistics* **5** (2017), 135–146.

[36] Q.V. Le and T. Mikolov, Distributed Representations of Sentences and Documents, in: *International Conference on Machine Learning*, 2014, pp. 1188–1196.

[37] H. Gómez-Adorno, J.-P. Posadas-Durán, G. Sidorov and D. Pinto, Document embeddings learned on various types of n-grams for cross-topic authorship attribution, *Computing* (2018), 1–16.

[38] M.D. Zeiler, ADADELTA: An Adaptive Learning Rate Method, *arXiv preprint arXiv:1212.5701* (2012).

[39] J. Finkel, T. Grenager and C. Manning, Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling, in: *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics*, ACL'05, 2005, pp. 363–370.

[40] I. Markov, H. Gómez-Adorno and G. Sidorov, Language- and Subtask-Dependent Feature Selection and Classifier Parameter Tuning for Author Profiling, in: *CLEF 2017 Evaluation Labs and Workshop – Working Notes Papers, 11-14 September, Dublin, Ireland*, CEUR-WS.org, 2017.