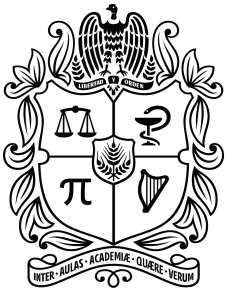

UNIVERSIDAD
NACIONAL
DE COLOMBIA

Automatic authorship analysis using deep neural networks

Sebastian Ernesto Sierra Loaiza

Universidad Nacional de Colombia
Engineering School, Systems and Industrial Engineering Departament
Bogotá, Colombia
2018

Automatic authorship analysis using deep neural networks

Sebastian Ernesto Sierra Loaiza

Thesis submitted as requirement to obtain the title of:
Magister in Systems and Computer Engineering

Advisor:
Fabio A. González, Ph.D.

Research Field:
Machine learning
Research group:
MindLab

Universidad Nacional de Colombia
Engineering School, Systems and Industrial Engineering Departament
Bogotá, Colombia
2018

Automatic authorship analysis using deep neural networks

Sebastian Ernesto Sierra Loaiza

To those who believe in the peace in Colombia.
To my family and friends, specially to my mother,
for her love and encouragement.

Acknowledgements

First of all, I would like to express my gratitude to my advisor, Professor Fabio Augusto González Osorio, whose support and excellent advice was always encouraging for completing my Master thesis. I am deeply grateful for all the academic and personal teachings. I owe special gratitude to Professor Manuel Montes, for having provided great advice and giving me the opportunity to have an internship at INAOE in Mexico. To Professor Thamar Solorio, my most sincere appreciation for the insightful comments and the encouraging work on authorship attribution.

To all my colleagues in MindLab laboratory, I want to express my gratitude for each advice, coffee and beer we shared. Special thanks to John Arévalo, Prasha Shrestha, Camilo Concha, Javier Ariza, Óscar Perdomo and Andres Rosso. To my friends, I feel lucky for having your invaluable counsel and friendship.

And most importantly, to my parents, Magda Loaiza and Carlos Sierra, my most wholehearted gratitude for your love and support during all these years. To my sister and my grandparents, thank you for your love and company.

I also want to acknowledge Colciencias for funding partially my research through the research projects FP44842-576-2014 and 1225-569-34920. Each of the people I have mentioned here, have contributed to complete successfully my master. Also they have helped me to define what I am. To all of them, my most sincere appreciation.

Abstract

Authorship analysis helps to study the characteristics that distinguish how two different persons write. Writing style can be extracted in several ways, like using bag of words strategies or handcrafted features. However, with the growing of Internet, we have been able to witness an increase in the amount of user generated data in social networks like Facebook or Twitter. There is an increasing need in generating automatic methods capable of analyzing the style of a document for tasks like: determining the age of the author, determining the gender of the author, determining the authorship of the document given a set of possible authors, etc. Previous tasks are better known as author profiling and authorship attribution. Although capturing the style of an author can be a challenging task, in this thesis we explore representation learning strategies, in order to take advantage of the large amount of data generated by social media.

In this thesis, we learned proper representations for the text inputs that were able to learn such patterns that are only distinguishable to an author (authorship attribution) or a social group of authors (author profiling). Proposed methods were compared using different publicly available datasets using social media data. Both author profiling and authorship attribution tasks are addressed using representation learning techniques such as convolutional neural networks and gated multimodal units. Our unimodal author profiling approach was submitted to the profiling shared task of the laboratory on digital forensics and stylometry(PAN).

For authorship attribution, we proposed a convolutional neural network using character n-grams as input. We found that our approach outperformed standard attribution based methods as well as word based convolutional neural networks. For the author profiling task, we proposed one convolutional neural network for unimodal author profiling and adapted a gated multimodal unit for multimodal author profiling. The multimodal nature of user generated content consists of a scenario where the social group of an author can be determined not only using his/her written texts but using also the images that the user shared across the social networks. Gated multimodal units outperformed standard information fusion strategies: early and late fusion.

Keywords: Machine learning, Supervised Learning, Representation Learning, Automatic Authorship Analysis, Authorship Attribution, Author Profiling, Multimodal Author Profiling

Contents

List of Figures

List of Tables

1 Introduction

Nowadays, social networks generate a significant volume of information. This volume is derived from the number of interactions between users in such networks. Images, text, and videos are examples of the elements that users post. The scientific community is using this large amount of information to generate automatic analysis methods that allow to characterize the information generated by users. For example, a twitter user can generate a variety of information describing her preferences, interactions with other users and opinions on specific topics. Website review portals such as Internet Movie Database (IMDB) are another source of information. IMDB offers a platform that allows users to write their review about a movie. This review describes positive, negative or neutral opinion of a movie. Although it does not generate the same amount of information as Twitter, it generates large amounts of text written by real users. Furthermore, the development of automatic methods for text analysis in large volumes of data can create and support different business models and applications.

1.1 Problem definition

Authorship analysis is the process of studying the characteristics of a document written by an author, in order to identify features associated with its authorship. Authorship analysis can also be divided into several subtasks.

One of them is authorship attribution (AA), which consists of automatically identifying the authorship of a new document of unknown or disputed authorship. Each document is represented using a set of features and the authorship is determined using computational learning methods. The problem of AA can be divided into two types: open-set and closed-set. In the closed-set domain, the model evaluates the authorship of a new document from a predetermined set of authors, where the evaluated document was written by one of the pre-established authors. The open-set tasks establish that the author of the unseen document is not necessarily found in the list of candidate authors.

Another task in authorship analysis is author profiling (AP), which studies the use of language across several demographic groups. AP is based on the hypothesis that social groups (profiles) share the use of language. Unlike authorship attribution, the target classes consist of demographic groups, which can be based on the author's gender, age, origin country, personality traits, among others. The process of extracting characteristics is very similar to the AA's process.

Feature extraction process in authorship analysis has been approached originally using techniques of stylometry. Stylometry aims to measure certain textual characteristics of the writings of each author. Overall, traditional methods for authorship analysis have focused on building representations using stylistic and content based features. Stylistic features attempt to model phenomena such as use of punctuation marks, average length of a sentence, use or misuse of grammar rules,

spelling mistakes and use of emojis. Content features attempt to capture the topics that an user writes about. Both types of features can be extracted using bag of n-grams representations.

Although these features have shown competitive performance in authorship analysis, they have also several shortcomings. For instance, when we build a bag of character n-grams representation for a document, we treat each document as a simple sequence of characters, then the frequency of occurrence for each n-gram is calculated. This representation poses three problems:

- This representation is very dependent on the vocabulary (total number of n-grams in all documents), so it can not scale easily. Many machine learning algorithms can not scale and work properly for solving the task while dealing with high dimensionality inputs.
- Each document will only contain a few words from the total vocabulary. This means, the bag of n-grams representation of each document will only have some values different than zero. This problem consists of sparsity, where many elements of the input will be zero. Sparsity affects certain learning algorithms which rely on heavy parameter updates and therefore will tend to overfit.
- The third problem is the inability to capture relationships between n-grams, since the bag of n-grams representation simply captures the presence or absence of a certain n-gram, therefore, two documents with a similar semantic meaning would have totally different representations using bag of n-grams. Additionally, Bag of n-grams ignores the sequence order of n-grams.

For improving document representation we will use representation learning based strategies. Representation learning offers the opportunity to build distributed representations that solve the problem of high dimensionality, sparsity and the inability to capture relationships between documents. Learned representations using neural networks are also considered distributed, which means several features in the representation can coexist and are not mutually exclusive. Also these representations have the additional advantage that they can be adapted perfectly to the supervised task that is being solved, either authorship attribution or author profiling. Approaching these tasks using neural networks involves solving three subproblems: first, to choose an *appropriate input representation* of the text; second, to determine if neural networks based approaches perform better than traditional methods; and third, what text inputs are more important for the neural model.

- How to learn an appropriate representation for authorship analysis tasks?
- Does this representation improves the use of traditional methods for authorship analysis?
- What kind of features are more important for the text classification method?

1.2 Objectives

Main objective To develop a method for automatic authorship analysis using deep neural networks.

Specific objectives

- To design a deep neural network architecture for automatically identify author of a text (authorship attribution).
- To design a deep neural network architecture for automatically characterize the author of a text (author profiling).
- To build efficient implementation of the deep neural network models that take advantage of acceleration hardware (GPUs).
- To evaluate the models in different authorship attribution and author profiling tasks.

1.3 Results and contributions

The results and contributions of this work can be summarized as follows:

- Shrestha, P., **Sierra, S.**, Gonzalez, F.A., Montes, M, Rosso, P., Solorio, T.. "Convolutional Neural Networks for Authorship Attribution of Short Texts". *In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers* (2017)

  In this work, we propose a convolutional neural network for the authorship attribution task. Our method is compared against standard attribution methods like bag of words and it is also compared against different kinds of text representation strategies. My contributions in this work include participation in the development of the code, design and execution of the experiments, writing of the draft and camera ready versions for the EACL 2017 conference and poster presentation for the conference.

- **Sierra, S.**, Montes, M., Solorio, T., Gonzalez, F.A.. "Convolutional Neural Networks for Author Profiling". *In: Working Notes Papers of the CLEF* (2017)

  In this work, we propose a convolutional neural network for the author profiling task. Our method is compared against standard profiling methods and it is also compared against different kinds of text representation strategies. My contributions in this work include the development of the code, design and execution of the experiments, writing of the draft and camera ready versions for the CLEF conference. Also this work was part of an evaluation lab named as PAN, which evaluates tasks in digital forensics like author profiling. Our work was in the top-8 methods among more than 22 participating teams.

- Shrestha, P., **Sierra, S.**, Montes, M, Rosso, P., Solorio, T., Gonzalez, F.A.. "A New Multi-channel Convolutional Neural Network for Authorship Attribution of Social Media Data". *In: Information Processing and Management* (under revision)

  In this work, we propose a multi-channel convolutional neural network for the authorship attribution task. Our method is compared against standard attribution methods like bag of words and it is also compared against different kinds of text representation strategies. My contributions in this work include participation in the development of the code, design and execution of the experiments and writing.

1.4 Outline

The document is organized in five chapters. Chapter 1 presents the problem definition, the main and specific objectives and the contributions of this work. Chapter 2 presents an overview of the state of the art for text representation techniques, authorship attribution and author profiling. In Chapter 3, a convolutional neural network for solving the authorship attribution task is presented. Chapter 4 presents a convolutional neural network for author profiling and an extension of this work to a multimodal author profiling scenario. Finally, Chapter 5 outlines some of the concluding remarks and future work.

2 Background and related work

The classification of texts is a task that consists of assigning a category or label to a document. To be able to do classification, the document must first be represented through a series of characteristics, which will then be used as input for machine learning methods. In recent years, the representation of text has become an important component of natural language processing, where it has been chosen to acquire or learn the meaning of a word using computational methods [1]. LeCun, Bengio, and Hinton [2] shows the distributed nature of word representations learned with deep neural networks as. Distributed representations have a greater expression capability, since they can express multiple concepts at the same time and their many configurations are a result from the variation observed from the data. Later in the chapter, there will be exposed different ways to represent a text, but the center of this background are the representations learned by neural networks. These representations are obtained through language models trained with neural networks. In this section we will show what a language model consists of and how the training of these models has evolved. At the same time, some of the characteristics that show the vector spaces induced by these models will be shown. Subsequently, some of the most recent works that address the problem of semantic compositionality will be shown. Finally, it will be shown how the use of representations learned by neural networks can help concrete tasks of text classification.

2.1 Text representation

2.1.1 Language Modeling

The distributional hypothesis states that words that occur in similar contexts tend to have a similar meaning [3]. Using this hypothesis, language models extract information from the semantic relationships between words contained in a large corpus. Language models consist essentially of the probability of occurrence of a particular sequence of words. That is, in a language model, the string "the cat sits at the table" has a higher probability than the string "the cat sits on the table". Usually the probability of a sequence of words $(w_1, \ldots, w_n)$ is represented as $P(w_1, \ldots, w_n)$ and is modeled according to its context, i.e., the words before and after the sequence.

$$P(w_1, \ldots, w_T) = \prod_{i=1}^{T} P(w_i | w_1 \ldots, w_{i-1}) \sim \prod_{i=1}^{T} P(w_i | w_{i-(n-1)} \ldots, w_{i-1})$$

Neural networks are machine learning tools that allow us to learn language models in tandem with a vector representation for words [4]. Because these spaces have been constructed taking into account the distributional hypothesis, it is possible to affirm that the distance between two vectors represents the semantic similarity between two words [5]. In addition, the vector spaces

induced by these models present a low dimensionality compared with other strategies of textual representation that can have dimensions of the size of the corpus vocabulary. This problem is known as the "*curse of dimensionality*" and that is what made it difficult to address the problem of learning a language model [4]. In the field of neural networks, this problem has been addressed using multilayer perceptrons and recurrent networks. [4, 6] represented the first learning method of large-scale language models using neural networks. In this work, they implemented a neural network architecture, using the most common type of neural networks for modeling the language. This architecture is known as multilayer perceptron or "feedforward neural networks". This neural network had an input layer, a hidden layer and an output layer. This network had as its basic idea to learn at the same time a representation of the words in a vector space of low dimensionality and the parameters of the joint probability function that would express the probability of a sequence of words. This probability was calculated in the output layer with an algorithm known as softmax which enables to establish a well-defined multinomial distribution between classes, i.e., words. Despite its advantages, the model is computationally expensive. This work also shows the capacity of the neural networks to learn distributed representations that help to solve the problem of high dimensionality.

The main advantage of this kind of representation is how neural networks assign cell activation patterns to different inputs, with the idea that similar entries will have similar activation patterns [7]. Several algorithms for learning in natural language processing take advantage of these distributed representations because they are allowed to group similar concepts. The training of neural networks is usually done through "Backpropagation", an algorithm which consists of adjusting the weights of the connections between layers of the neural network trying to reduce the prediction error of the output layer of the network. It is also considered an optimization problem where we try to minimize the error in the output, for which an algorithm known as stochastic descent gradient (SGD) is commonly used. SGD allows to estimate how much the weights of the connections between layers have to be modified. In the end, we expect that certain cells in the intermediate layers are activated in the presence of a certain input pattern.

On the other hand, recurrent neural networks have recently been proposed to address the issue of language modeling [8, 9]. Feedforward neural networks always receive a fixed-size entry, i.e., they can only take into account a determined number of fixed words that constitute the context of a word. The advantage of recurrent networks is that they can process inputs of any size in a sequential manner, since they have recurrent connections that enable the network to learn long-term relationships. Although it has been established that some methods like stochastic descending gradient have difficulties to learn long-term patterns [10]. Recurrent neural networks consist of an input layer, a hidden or context layer and an output layer. The difference with multilayer perceptrons is that this layer of context has access in a time t to the state of the same context layer at time $t-1$. This is also known as recurrent connection. Training in recurrent networks is performed using "Backpropagation through the time" (BPTT). BPTT calculates the error and propagates it through the recurrent connections a certain amount of steps back in time. In the last layer, the probability distribution over the word sequence is calculated, where the number of possible classes is equal to the size of the corpus vocabulary, which makes the task of calculating this distribution complicated. This calculation is carried out using a softmax type classifier in both recurrent networks and in multilayer perceptrons.

Despite this, softmax is computationally expensive when carried out over a large number of classes, so it has been proposed for this particular case the use of a hierarchical softmax (HS) [11]. HS uses an underneath representation of the words through a binary tree. This simplification of softmax has also been used for the training of multilayer perceptron networks [12]. Mnih and Teh [13] address the problem of computational cost making a smarter sampling of the words it chooses to train through a technique known as contrastive noise. These distributed representations are learned in an unsupervised fashion with the goal of creating general purpose representations for text classification tasks. Once the language models have been trained, they can also be used as extractors of text features for semantic classification tasks. One advantage of the learned vectors with neural language models is that these distributed representations are very useful in different natural language processing tasks [14]. The intention of this work was to learn these representations through multitasking learning. The tasks on which the model was trained were: part-of-speech tagging, named entity recognition, text partitioning, language modeling and synonymy detection. This work also showed that it was not necessary to extract syntactic information from the text to obtain semantic information. Subsequently, the same authors, in collaboration with other researchers, formulated a neural network architecture [15] that can be trained on large volumes of non-annotated text, in order to learn robust representations of text. These representations were then independently tested in tasks such as part-of-speech tagging, text partitioning, semantic role labeling and named entity recognition.

Later in 2013, Mikolov et al. [12, 16] addressed the computational cost issue using a network of feedforward neural network without hidden layers. Having a reduced number of layers also reduces the number of parameters that had to be estimated during the training and therefore reduces the computational complexity. This language model is known as **word2vec**. The training of this model can be done with the idea described above of hierarchical softmax or also using negative sampling. In addition to the computational efficiency that word2vec offers, this models learns an embedded space that presents certain similarities between pairs of words, also named linguistic regularities [17] by Mikolov et al.. These linguistic regularities can be semantic or syntactic and are used as an evaluation measure to determine if a model is capable of capturing singular / plural relationships such as: car: cars; dog: dogs; country: capital; Germany: Berlin; France: Paris. Interestingly these relationships are found by simply using arithmetic on the vectors, for example, the word vector for "Germany" will be very close to the result of the result of the vector operation between the following words: vector('France') - vector ('Paris') + vector ('Berlin'). However, it has been shown that these regularities can be obtained in embedded spaces, not only by neural networks but by distributional representations in general [18]. To measure how well a language model captured these linguistic regularities, an evaluation strategy was found in the literature, also known as an analogy problem [16]. It consists in determining the missing word in the analogy "if a is to b as c is to _", where the missing space is the word that the model must be able to predict. Analogies can be evaluated at syntactic and semantic relationships between words [17].

2.1.2 Semantic compositionality

From the field of computational linguistics, these language models have been positioned as a tool for learning the meaning of words. There is an interest in this problem because the ability to learn

good representations of sentences, paragraphs or even documents can help to solve specific tasks of text classification such as sentiment analysis, paraphrase detection, machine translation, among others. This problem is known as semantic compositionality. Some works have focused on learning strategies to combine two words and form expressions [19–21]. These works focused on learning simple vector composition functions such as addition, substraction and element-wise multiplication. However, the literature suggests that the relationships between words must be learned jointly with the word representation training process. Le and Mikolov [22] propose an interesting line of work, which consists of integrating the learning of language models in neural networks with the learning of representations at the sentence level. In this work the word2vec model is extended and they proposed to learn the sentence's representation as an additional word during the training process (this method has been known as '**doc2vec**'), however, it lacks of a concrete method to infer unseen sentences during training.

Li, Luong, and Jurafsky [23] use a strategy with recurrent networks as auto-encoders. This work proposes using an encoder in the first instance to build an embedded representation of the sentences and then, through a decoder, to use this representation for reconstructing the original sentence. Kiros et al. [24] follow a similar approach known as Skip-Thoughts, in which a recurrent network is used to train a model that is capable of representing sentences. The key to the training of this method is to use an encoder-decoder architecture. This strategy consists of first choosing a tuple of 3 joint sentences (s_{t-1}, s_t, s_{t+1}), then sentence (s_t) is represented using an encoder, and its representation is used for reconstructing context sentences (s_{t-1}, s_{t+1}) using two decoders. The advantage of this encoder is that it serves as a generic feature extractor for any sentence, as the authors demonstrate in concrete tasks of semantic similarity such as detection of similarity between sentences, paraphrasing detection and textual classification. These works motivate the idea of incorporating the learning of relationships between words that make up the meaning of a sentence to the training of a language model.

2.1.3 Text classification

One of the interests of this thesis is to apply the learned representations through neural networks to a task of text classification, concretely, authorship attribution and author profiling. As shown previously, neural networks have demonstrated the ability to induce a space where a measure of similarity between words can be established. Next, we will name some tasks that take advantage of the notion of semantic similarity as a discriminating element between classes.

Paraphrase detection is a well-known task in the field of semantic similarity. It consists in determining if two sentences are describing the same idea. Dolan, Quirk, and Brockett [25] released the so called dataset "Microsoft Research Paraphrase Corpus", which consists of a total of 4076 pairs of training sentences and 1725 test pairs. The state of the art strategy was established by Madnani, Tetreault, and Chodorow [26], whom work uses 8 metrics from machine translation domain (MTMETRICS) and overall it takes advantage of several external resources to the dataset. On the other hand, there is another set of paraphrasing data on Twitter [27] that contains approximately 18,000 couples of paraphrasing (or not paraphrasing) in training and 1000 couples in test. Also in [27], the best state of the art strategy is reported, which uses multi-instance learning about different extracted lexical characteristics about the words of each sentence, such as topics,

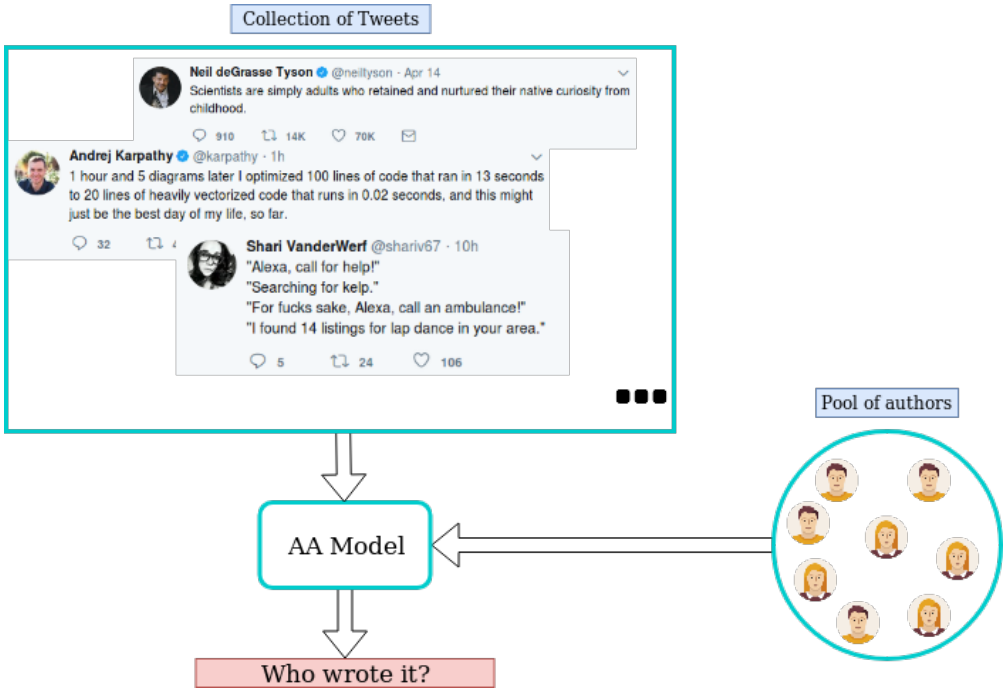

Figure 2.1: Closed-set case for authorship attribution. The task consists in determining the author of an unseen document.

part-of-speech tags and cardinality. The key to this strategy is to learn the patterns of relationship between these characteristics together, in order to recognize which patterns make a couple of sentences a paraphrase. The task of paraphrasing has also been related to the detection of plagiarism. The detection of plagiarism is to identify when an author uses the ideas of someone else without recognizing the contribution made by the other person. In [28], they proposed to formulate the detection of plagiarism as a problem of paraphrasing, with the intention of identifying not only if there is plagiarism, but also to identify the kind of plagiarism that occurs. The identification of key elements to model semantic similarity in text classification can also help to solve other problems that do not require predicting a category but a numeric value. One of these problems is known as semantic similarity between sentences and it consists of determining on a numerical scale how similar two sentences are. The dataset that has gained popularity for this strategy is known as SICK dataset [29]. Currently the best strategy that addresses this task is called Tree-LSTM [30]. In this work, recurrent networks and syntax trees are used to determine the similarity between two sentences.

2.2 Authorship analysis

2.2.1 Authorship attribution

In authorship attribution (AA), we are interested in determining the authorship of an unseen document. AA is not a new problem. Mendenhall [31] explored several features for characterizing the works of Shakespeare in the 19th century and determined that it was a difficult task to solve

for documents with less than 1.000 words. Although, AA can be seen as a text categorization task, it has a wide variety of evaluation scenarios that make it different from typical text categorization tasks. According to Stamatatos [32], authorship techniques shifted to computational-based methods rather than computational-assisted methods. AA process consisted of extracting hand-crafted features like word and character frequencies, sentence length, word length, among others.At the beginning of the 21th century, AA had a great boost derived from the advances in other related fields. Advances in information retrieval field allowed the use of text representation techniques for classifying large volumes of text, while natural language processing (NLP) techniques allowed the extraction of some more advanced representation like Part-Of-Speech (POS) tags [32].

One of the focus of authorship works has been attribution of long texts such as books [33–35]. In the last years, there has been an increasing amount of available text from social networks, therefore AA has shifted towards the study of short texts such as tweets [36, 37], blogs [38] or emails [39]. According to Stamatatos [32], performing AA on short texts also depends on several factors: the number of candidate authors, the text size and number of available texts for training. Performing AA involves a feature extraction process and an algorithm training process. Feature extraction process include a wide range of features such as lexical, syntactic and application-based features [32]. Lexical features can be word and character n-grams. Syntactic features like Part-of-Speech (POS) tags are also used for authorship attribution. Application-based features have into account the use of topic models that capture topics in the documents [40–42]. Additional features can be extracted from a document using measures like complexity measures, the readability index, among others. Most of the representation strategies lead to a high dimensionality representation space. As can be seen on [33], a dimensionality reduction or a feature selection step must be performed in order to apply consistently machine learning algorithms that have a good performance but may not scale properly.

Most authorship attribution research make use of character n-grams and/or word-level token n-grams in one way or the other [32, 36, 37]. Character n-grams are powerful indicators of the authorship of a document by capturing the style of an author. Character n-grams also are robust to typos and non-correct use of punctuation marks Stamatatos [32]. These features usually include whitespaces, in order to capture relationships between several words. According to Sapkota et al. [43], character n-grams can be sub-categorized on the basis of morphology: affix n-grams, punctuation n-grams and word-based n-grams. Affix n-grams are found at the beginning or at the end of a word (prefixes and suffixes). Punctuation n-grams involve sequences of characters that involve punctuation marks. They found also that affixes and punctuation category of n-grams were the best for authorship attribution. For instance, punctuation n-grams capture the misuse of punctuation marks, that is, while some authors prefer to add only one exclamation mark at the end of a sentence, some other prefer to add two, three o more. This represents an author's style and character n-grams are ideal at capturing this. Similarly, the choice of some words is also mostly unique to an author. Lexical features can help to distinguish between different authors based on the conscious or unconscious choices they make. Some researchers [32, 44] suggest that smaller values of n sequential characters result in better performance in authorship tasks. Stamatatos [45] also use character n-grams as features along with dealing with the problem of class imbalance for authorship attribution by trying various sampling methods. Lexical-based methods are a strong baseline to compare with.

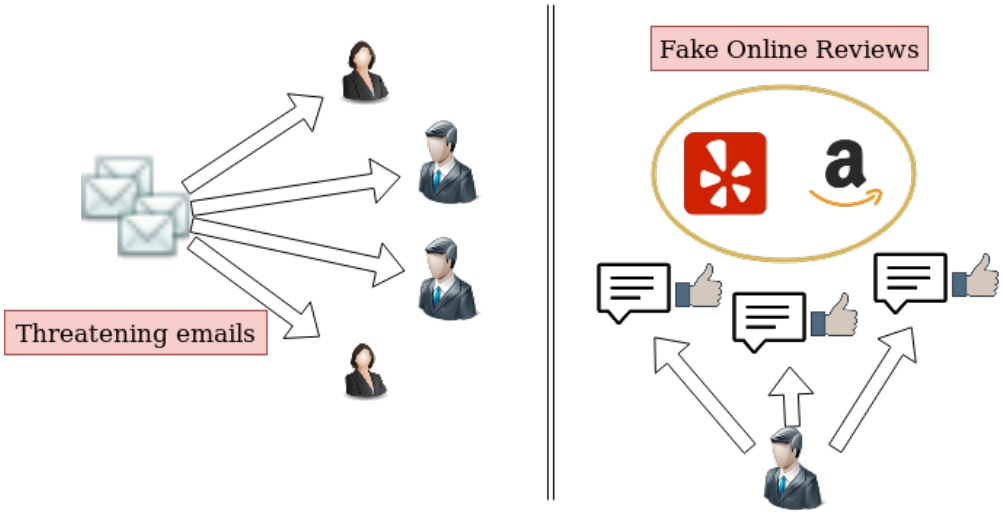

Figure 2.2: Applications of authorship attribution in social media texts.

Authorship attribution of social media texts

The application of authorship attribution to short texts has been motivated by the massive use of social networks. Social networks like Twitter allow the user to generate text of up to 280 characters (previously it was up to 140), which motivates the creation of attribution methods oriented to short texts. The generation of methods capable of evaluating authorship in social networks such as Twitter, raises the need to establish a preprocessing strategy. Although a tweet is considered a self-contained document, it has peculiar features such as references to other users, hashtags, urls, spelling errors, slangs and abbreviations. Applications to AA in short texts have to deal with the reduced length of the document, to deal with scenarios where the number of authors is very large and additionally authors can write about a wide variety of topics. Below we describe recent works that

At the Twitter level, Layton, Watters, and Dazeley [37] collected a data set of $14,000$ users. Then, using the SCAP [46] method, each user was represented using profile consisting of bag of character n-grams representation. Characters from 2 to 7-grams were explored. They found that 4-grams generated the best performance for determining authorship on Twitter. They additionally evaluated the effect of hashtags and usernames preprocessing, finding that if the usernames and hashtags are kept, the classification accuracy is higher. This is a result from the fact that some users have constant conversations with certain Twitter accounts, where the tweet includes itself the mention to the other accounts, therefore the n-grams that constitute the name of an account will appear more frequently in the profile representation of each user. In another work, Schwartz et al. [36] propose a method of authorship attribution in short texts. This method is evaluated in two setups: varying the number of training instances (number of tweets per author), and varying the number of authors. For the feature extraction process, they explore the use of character n-grams and word n-grams. For characters, they consider 4-grams, meanwhile at word level they consider from 2 to 5-grams. Each document is represented as a whole by the occurrence or not of a certain n-gram. Then, using k-signatures, they represent the presence of a certain characteristic in the

k% of an author's text and the absence in the rest of the dataset. Besides the character and word n-grams, they build features using flexible patterns, which are defined by a sequence of content words (words that contain information) and functional words (high frequency words like articles that connect content words). Despite the fact that their results are competitive, they have to define manually some signature templates for the different authors.

In a similar sense, Iqbal et al. [39] propose a strategy to determine the authorship of an email through write-prints. These prints are composed of different patterns observed in the writing of a document. Such patterns are extracted with stylometric measurements. One of the applications they propose is to present evidence before a court to determine whether or not a document was written by a person, who is for instance, accused for sending threatening mails.

In review pages like Amazon, the scenario is even more challenging. In many cases there is a reduced number of opinions per person. In this scenario, Qian et al. [47] proposes a scenario of attribution of authorship in which, instead of training a classifier that separates the documents of one author from those of another, it builds a similarity space of documents. In this new similarity space, it obtains competitive performance by using only a few couple of documents from an author during the training stage. They also present the scenario in which this method is useful to identify users who create false accounts and fraudulently promote the opinion about a product. This scenario can be seen on Figure 2.2.

Most of these traditional methods focus on crafting features and statistics based on lexical choices. In some other cases, there is need of manually selecting which features are relevant for addressing the attribution task. Traditional methods have shown the importance of character n-gram input representations, but we can connect deep learning techniques for solving the attribution task, while learning a representation for the input text.

Neural representations for authorship attribution

Deep learning representations have achieved outstanding results in several natural language processing tasks LeCun, Bengio, and Hinton [2] and Mikolov et al. [12]. Authorship attribution has started to apply some of the deep learning architectures that have been used successfully for text categorization. Text categorization has been addressed by Convolutional Neural Networks (CNNs). This type of neural networks has also shown state-of-the-art performance in various Computer vision tasks. CNNs applied for text have been used to represent sentences or paragraphs [48–51]. These works use as input either a sequence of words or a sequence of characters. As will be shown in this work, CNNs can learn local patterns like the use of certain combination of emojis using character level inputs. Recent works have also used CNN at the character level. Kim et al. [52] proposed a neural language model that combines a CNN with an RNN through a highway network. Zhang, Zhao, and LeCun [50] trained a neural representation CNN architecture and showed that these networks outperform traditional approaches on many document classification tasks using large datasets.

CNNs have been used for authorship attribution only in a few works. Rhodes [53] uses CNN for authorship attribution using a dataset of books from the Project Gutenberg and another dataset of books from the PAN 2012 shared task [54]. He achieves a high accuracy percentage (85.7%) using word representations with CNNs and a voting schema for summarizing the predictions of

each sample of the books. In other work, Sari, Vlachos, and Stevenson [55] use fastText for training an authorship attribution model. They use the Bag of tricks model [56], which trains a distributed representation for word and character n-grams while solving the attribution task. They try character 2,3,4-grams and word 1,2-grams separately as well as combined on four different datasets and are able to obtain good performance on the CCAT50 and IMDb62 datasets. It is worth to mention that n-gram representation was learned along with the supervised task using fastText. Embeddings from fastText can also be used to initialize the input layer of a CNN.

Multi-channel models for text classification

Multi-channel models, in the context of CNNs, consist of a network with more than one channel, where each of them will be receiving the same input. Either the representations of the text itself differs in these channels or the structure and training methods themselves vary for each of these channels. There have been a few multi-channel models that have been tested on text classification tasks such as sentiment analysis and subjectivity classification [49, 57]. There are two most common ways in which multi-channel CNN models are used for text classification in general. First way is centered around keeping one channel static, while fine tuning pre-trained embeddings in the other channel [49, 58]. The second method uses different forms of pre-trained embeddings in the different channels [57].

Apart from these two ways, there is a newer way of using multi-channel models. Here the various channels get entirely different representations of the input text. We came across only one such work by Ruder, Ghaffari, and Breslin [58], where they try different single-channel and multi-channel CNNs, with a combination of static and non-static channels for word sequences and character sequences. The embedding layer is retrained after initialization for the non-static channel whereas the embedding layer is not updated after initialization for a static channel.

For their multi-channel models, they combine the feature representations obtained after the convolutional layer and before max pooling. Since the feature vector from the character channel will be longer than that from the word channel for a certain filter size, they pad the word feature vector (padding is usually done with zeros) to make it equal to length the of the character vector and then add the two vectors in order to obtain the final representation, which is then passed to a dense layer for authorship attribution. They were unable to obtain good performance over single-channel models with their methods for the majority of their experiments.

2.2.2 Author Profiling

Author profiling consists of determining a social group of an unknown author [59]. Chambers and Schilling [60] support this idea with a sociolinguistics observation, where a social group shares a way of speaking and writing, a dialect. Several profile dimensions for characterizing a social group have been considered since then, such as age [61], gender [62], native language [63] and personality. The relevance of this task has been recognized for its applications that include forensics, marketing and security concerns.

Author profiling has been addressed in the CLEF conference, under the PAN shared task. PAN is an evaluation laboratory for digital forensic tasks like plagiarism, authorship and social software misuse [64]. Gender detection is one of the most popular subtask in author profiling [61, 65–71].

This task has been approached using two kinds of features, style-based features and content-based features. Style-based features included n-gram frequencies, punctuations, readability. Whereas content-based features comprise bag of words, word n-grams, term vectors, named entities, among others. Previous successful approaches have used style and content features. Argamon et al. [59] showed experimentally that content features performed better for language, age and gender profiling. However these features can be very sparse. López-Monroy et al. [72] approached the profiling task using a low-dimensional non-sparse representation of the documents of every author. Other studies even describe that not all words matter when establishing the profile of an author, but suggest that words near a personal pronoun are more discriminative for classifying an author's profile [73].

CNNs have proven to be a successful method for classification of texts [48–50]. CNNs have also shown a good performance on authorship attribution tasks [74]. CNNs are suitable for the author profiling task, given that they are capable of capturing local-level interactions while learning profile-specific patterns.

Multimodal author profiling

Nowadays social media has allowed us to communicate and share images, text, audio or video through the social networks. Although each social network tries to focus on different kinds of users, most of them allow the users to share their opinion towards an specific topic or to mention recent events in their life. This has lead to a huge availability of user-generated content. Although, social networks provide users with privacy options and identity protection, a large number of users make publicly available their demographic information, opinions, images, tweets or posts. On the other side, Author profiling has been recognized as a task using only text documents, however this task can also be addressed using images. There has been a recent interest in exploiting the multimodal nature of social media data, for instance, images, audio and text. Multimodal approaches use data fusion techniques, in order to combine different information sources [75]. Information fusion considers the problem of merging correctly two different representations of the same concept [75, 76].Atrey et al. [76] considers three levels of information fusion: feature level or early fusion, decision level or late fusion, and hybrid approaches. For this work, feature level consists of extracting text and visual representations and combining them into a single learning method. These combinations ignore the intrinsic correlation between modalities [77]. Decision level consists of combining the output decisions of previously learned classifiers for each modality. Hybrid approaches consist of methods that create a joint space for representing the different modalities of a concept, for instance for solving image captioning tasks [78, 79]. Multimodal approaches for author profiling have been considered by [80–82]. Álvarez-Carmona et al. [80] extend the PAN AP 2014 corpus by extracting a large set of tweets and images from the original users of this corpus. While their fusion strategy consists of an early fusion of text and image features. However most of them ignore relations that could arise from the multimodal nature of the data. Taniguchi et al. [82] propose a hybrid fusion strategy, where visual concepts are extracted using a CNN, but each concept has a probability of being associated to a dimension of the profile (male or female). Text representation is extracted as the probability of a document to belong to a female user or a male user. At the very end, all the probabilities are concatenated and fed to a logistic regression classifier. Using an integrated

representation can lead to a better performance, compared to standalone profiling methods. Merler, Liangliang Cao, and Smith [83] evaluate gender prediction based on a large collection of Twitter users. Early, late and custom fusion approaches are reported there.

Author profiling task has been approached as a single source task, where texts written by different authors and different modalities are presented. Gender identification based on the images that an user posts in his/her social media is a task that has been gaining interest [84–87]. Most of this works take the images of a social media user, extract the visual concepts, for instance, if a bag is present in the image and finally associate the presence of this concepts to the gender of the user (profile). Shigenaka, Tsuboshita, and Kato [85] interestingly propose a neural architecture which learns a proper representation for the images while associates it with the visual concepts which are extracted from the images. Multiple sources of information can be combined appropriately using multimodal representation techniques. Finding an appropriate combination of features is a difficult task.

3 Using neural networks for authorship attribution

In this chapter a neural network for authorship attribution for short texts is presented. Representation learning models based on neural networks attempt to automatically find data features useful to solve a learning problem. In the particular case of authorship attribution, stylistic features may be found at different levels (morphological, lexical and syntactical). It means that a model able to automatically capture features at all these different levels must start at the most simple level. Thus, the model proposed in this chapter has an architecture that receives as input a sequence of characters n-grams. The complete model is composed of three modules depicted in **3-2**: a character embedding module, a convolutional module and a fully connected module. The gradients of the network are collected using back-propagation. Figure 3.1 displays the proposed architecture. The architecture is a convolutional neural network that uses a sequence of character n-grams as input. Our method was evaluated using the dataset proposed by Schwartz et al. [36], where several setups were used to measure the performance of our method in short texts. At the end of the chapter, we evaluate qualitatively the results generated by our proposed method using interpretability techniques. Part of this work was presented for the 2017 edition of the conference of the european chapter of the association for computational linguistics (EACL 2017):

- Shrestha, P., **Sierra, S.**, Gonzalez, F.A., Montes, M, Rosso, P., Solorio, T.. "Convolutional Neural Networks for Authorship Attribution of Short Texts". *In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers* (2017)

3.1 Character based convolutional neural networks

This neural network takes a sequence of character n-grams as input. Then, for every possible n-gram in the corpus, a representation is learned using an embedding layer. The embedding layer is motivated by the previous work developed on distributed representations [2]. The embedding layer learns a continuous and non sparse of the vector representations of each character n-gram in the input. Each input is padded, in order to guarantee the same length for all the input samples. The application of the embedding layer yields a matrix $C \in \mathbb{R}^{d \times l}$, where the columns are the embedding of the character n-gram c_j at position j in the input. Also if an input is shorter than l, it is padded with zeros until every input is of size l. The next component of the architecture is a convolutional layer, which applies an one-dimension convolution over the matrix C. This convolution is generated by a parametrized filter $H \in \mathbb{R}^{d \times w}$, where w is the width of the filter. As can be seen on Figure 3.1, several filters of different widths are used in order to capture patterns that can involve from

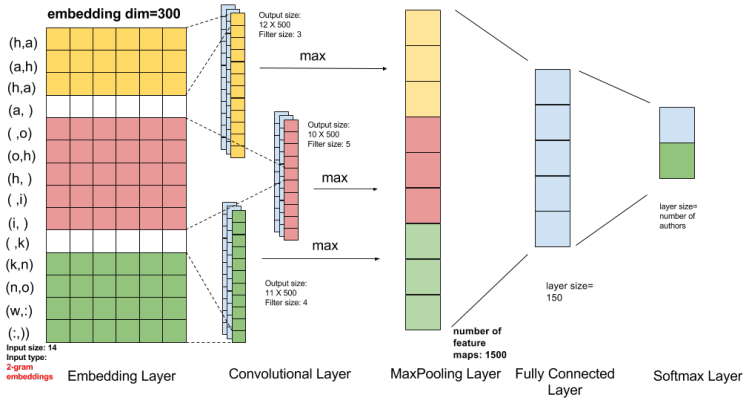

Figure 3.1: Char-level Convolutional Neural Network. In this concrete case the tweet input is **haha oh i know:)**.

morphemes up to words. The output C of applying a filter H to the input C is represented by:

$$O = H \cdot C[i : i + w - 1]$$

$$f = g(H \cdot C[i : i + w - 1] + b)$$

where i varies from $1 \dots l - w + 1$. Then, a bias is added and a non-linearity g is applied to the output O. This result f is also known as the feature map $f \in \mathbb{R}^{l-w+1}$. After that, a pooling function is applied in order to get a fixed representation of the feature maps extracted. Relevant features get higher activation values in their respective feature maps, so max-over-time pooling [51] is used. Thus for a m number of filters H the maximum value of each feature map f_k is extracted in the following way:

$$y_k = \max_i f_k[i]$$

where $k = 1 \dots m$. Overall three different convolutional layers are applied to the input and their results are pooled and concatenated, generating a vector of $3 \times m$ dimensions. Finally, this representation is fed to a fully connected module, which contains a Softmax layer, with a size depending on the number of target authors, which performs the final classification. Models using representation learning techniques have the advantage of finding automatically useful features that help to solve the classification problem. In authorship attribution domain, stylistic features are found across morphological, lexical and syntactic levels. This neural network is able to automatically capture these patterns starting by short sequences of characters and then using convolution to generate representations of longer sequences.

3.2 Experimental Evaluation

3.2.1 Dataset

We evaluated our approach on the dataset from Schwartz et al. [36] containing ∼9,000 Twitter users with up to 1,000 tweets each, using the same train/test splits, and normalized URLs, usernames,

CNN-2	CNN-1	SCH	CHAR	LSTM-2	CNN-W
0.761	0.757	0.712	0.703	0.645	0.548

Table **3-1**: Accuracy for 50 authors with 1000 tweets each.

and numbers. We trained separate CNN models with character n-grams ($n = 1, 2, 3$) on a small validation set. Here we evaluate our two best-performing models, one on unigrams (**CNN-1**) and another on bigrams (**CNN-2**), against three other systems described below:

SCH: The [36] work uses character 4-grams and word 2-5 grams. They also introduced k-signatures and flexible patterns to represent the unique signature of an author. Their best system uses a combination of all these features.

LSTM-2: Long Short Term Memory networks (LSTM) have been successfully used for text classification [88, 89]. We evaluate an LSTM trained on bigrams, since the LSTM produced better results on a small validation set.

CHAR: Character and word n-grams have been the core of many AA systems [36, 37, 90]. We tested various n-gram combinations on the small validation set and our final system uses character 2,3,4-grams with logistic regression.

CNN-W: Many works on CNN use word sequences as input [48, 53]. We also trained a CNN model with Google Word embeddings [12] fed to a static embedding layer.

All systems use cross-validation over the training set for hyperparameter tuning.

3.2.2 Results

We first experimented with a relatively small set of 50 authors and their 1000 tweets each. The results are in Table **3-1**. The results show that our CNN bigram model (CNN-2) performs very well on this dataset and outperforms the SCH system by nearly 5%. CNN-1 also exceeds the SCH method but is marginally worse than CNN-2, showing that there is merit in exploring the training of a CNN model on n-grams rather than only on single characters.

Layer	# of layers	Hyperparameters	
Embedding	1	l	140
		d	300
Convolutional	3	m	$[500, 500, 500]$
		w	$[3, 4, 5]$
		Pooling	max
Fully connected	1	# of units	Depends on the # of authors

Table **3-2**: Neural network architecture hyperparameters

Table **3-2** contains the combination of hyperparameters for the three modules that generate the best validation score. Additionally, we have added a dropout layer with 25% dropout after the first embedding layer for regularization. We then shuffle and group the samples into mini-batches of size 32 for faster training. We employ Adaptive Moment Estimation [91] with a learning rate of

$1e-4$ to train our network. We train for a maximum of 100 epochs and choose the model with the lowest validation error.

Varying number of authors and tweets

# of authors	CNN-2	CNN-1	SCH	CHAR	LSTM-2	CNN-W
100	0.506	**0.508**	0.425	0.412	0.338	0.241
200	**0.481**	0.473	0.411	0.409	0.335	0.208
500	**0.422**	0.417	0.355	0.342	0.298	0.161
1000	**0.365**	0.359	0.303	0.291	0.248	0.127

Table **3-3**: Accuracy comparison for increasing # of authors with 200 tweets per author.

We also wanted to explore how our method fares against the other methods when the problem becomes more difficult, i.e. when the number of authors increases or when the number of tweets per author decreases, as done in [36]. The results for increasing number of authors are shown in Table **3-3**. Both our CNN models perform fairly well above the other methods for all our experiments. Although the accuracy decreases with the increasing number of authors, even with 1000 authors our model obtains an accuracy well above 36%, and there is a 6% improvement over the state-of-the-art (SCH).

# of tweets	CNN-2	CNN-1	SCH	CHAR	LSTM-2	CNN-W
500	**0.724**	0.717	0.672	0.655	0.597	0.509
200	**0.665**	**0.665**	0.614	0.585	0.528	0.460
100	0.613	**0.617**	0.565	0.517	0.438	0.417
50	0.542	**0.562**	0.507	0.466	0.364	0.366

Table **3-4**: Accuracy comparison for decreasing # of tweets per author for 50 authors.

We can draw similar conclusions from the results where we decrease the number of tweets per author as shown in Table **3-4**. Following the work in SCH, these results are an average of the accuracy values obtained from 10 disjoint datasets. The performance of our system is fairly stable even when the number of tweets per author is low. The improvement margin actually increases slightly as we move towards a lower number of tweets.

A statistical t-test on the results over the 10 disjoint datasets shows that the difference between CNN-2 and CHAR, LSTM-2, and CNN-W are statistically significant at $p < 0.001$. We could not perform a test with SCH results as the individual disjoint dataset results are not reported. In both these tables, we can see that the CNN-2 model outperforms the CNN-1 model for experiments with more data points (higher no. of authors and/or tweets), which can be attributed to CNN-2 having a higher number of parameters to train. CNN-W performs worse than the other systems. Char-based inputs specialize on stylistic patterns whereas word-based ones focus on content-related patterns, which are less important for AA. This finding is consistent with previous research in AA [90, 92, 93].

CNN-2	CNN-1	CHAR	LSTM-2	CNN-W
0.683	0.678	0.609	0.525	0.420

Table **3-5**: Accuracy values for 35 authors with 1000 tweets each after bot-like authors removal (15 authors were bots).

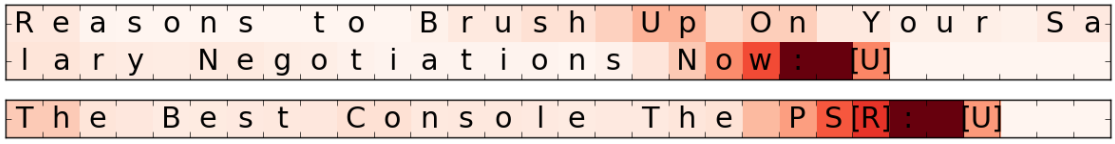

Figure 3.2: Salient sections of a bot-like author's tweets ([U]:URL, [N]:username, [R]:number).

3.2.3 Bot-like Authors

During analysis, we noticed that nearly 30% of authors behave like automated bots. Their tweets show repeated patterns, e.g., a title of some news/advertisements with a URL at the end. Since our goal is to perform AA on humans, we removed these authors manually to create a refined dataset. There are no comparable experiments in [36], thus we compare only against CHAR, LSTM-2, and CNN-W as shown in Table **3-5**. The accuracies for all of the methods decrease on this dataset as the bot-like authors are easy to identify. The CNN methods still outperform other methods. Since SCH's performance was similar to CHAR on the whole dataset and CNN-2 exceeds CHAR by a larger margin in this dataset, we can estimate that here too, CNN-2 is likely to outperform SCH.

3.3 Interpretability

Despite the competitive performance of neural representation techniques in several NLP tasks, there is a lack of understanding about exactly what these models are learning, or how the parameters relate to the input data. Few empirical studies have attempted to understand the role of RNN components [94, 95]. In order to analyze what makes neural representation learning suitable for AA, we look at the most salient sections of a single input tweet. We also perform an analysis of what types of character n-grams are more important to the model overall.

Salient sections of a tweet Li et al. [96] define a saliency score $S(e)$ as:

$$w(e) = \frac{\partial(S_c)}{\partial(e)} \qquad\qquad S(e) = |w(e)|$$

where the embedding e represents the input and the class score S_c represents the output of our CNN model. The score indicates how sensitive a model is to the changes in the embedding input, i.e. in our case, how much a specific n-gram in the text input contributes to the final decision. In order to visualize saliency per character, we adapted this method by taking the maximum saliency value per character.

We selected two authors, one bot-like and one human, to analyze what kind of patterns are learned for specific authors. Figure 3.2 presents two tweets from a bot author. The darker the

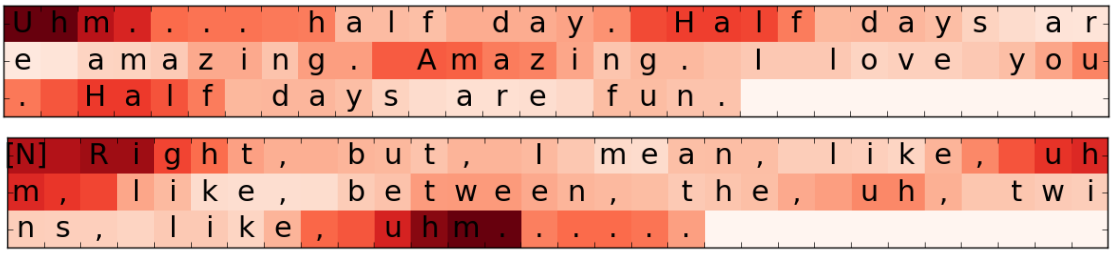

Figure 3.3: Salient sections of a human author's tweets ([U]:URL, [N]:username, [R]:number).

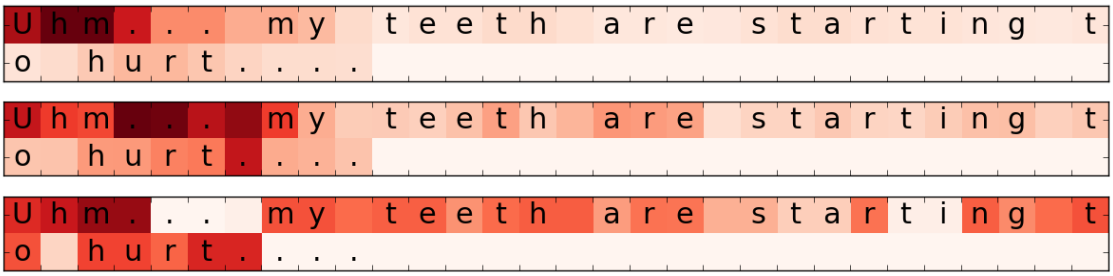

Figure 3.4: Salient sections comparison of CNN-2 (top) and CNN-1 (mid). The bottom figure is shaded using the feature weights from logistic regression for CHAR.

shade is, the more salient that section of the tweet is in the attribution decision. This automated bot seems to follow the pattern *Title: URL* and sure enough, it is detected by the CNN-2 model as indicated by dark shading towards the end of both tweets. Similarly, Figure 3.3 shows two tweets from a human author. We can notice right away that this author has the tendency to use *uhm* and we can see this section highlighted in the figure. The author also tends to use consecutive dots, this too is highlighted, albeit a little less than *uhm*. Figure 3.4 shows the saliency values for a tweet from the CNN-2 (top) and the CNN-1 (mid) models. For the CNN-1 model, although *uhm* and ... are highlighted, the saliency values are more distributed throughout the tweet, highlighting even *are* and *hurt*. While we can see that the CNN-2 model puts its focus exactly on the *uhm*, which is a very distinctive style of this author. Figure 3.4 also has a similar figure for the CHAR model at the bottom, which we created by using the feature weights from the logistic regression classifier. Although there is more focus on the *uhm* part, again, the distribution is more spread out for this model as well, compared to the CNN-2 model.

N-grams with highest contributions

Dataset	Highest activations overall
Bot & non-bot	_[U], Di, :l, n‘, (:, Xn, KM, o), :_, =h, _[R], :-, qh, wu, !,
Non-bot only	_[U], qh, KM, Di, (:, Uh, ;D, :p, _[N], __, :l, !_, =D, :_

Table **3-6**: Input char bigrams with highest CNN activations overall ([U]:URL, [N]:username, [R]:number, _:whitespace).

Some n-grams activate several filters, but generate low activation values, meanwhile, other n-

Dataset	Top activations per filter
Bot & non-bot	bi, ul, al, ug, me, in, mp, AN, um, an, en, "w, sa, e_
Non-bot only	_t, _m, er, ou, e_, in, ed, co, _a, is, nd, _r, ve, te, st

Table **3-7**: Input char bigrams with highest CNN activations per filter ([U]:URL, [N]:username, [R]:number, _:whitespace).

Dataset	CHAR top features
Bot & non-bot	:_[U], :, _-, _u, _r, .._-, _X, XD, _XD, li, go,_., _#
Non-bot only	...;-), lol, :d, maoo, &&, :)), :-(, :-p, loll, ????, ^_^

Table **3-8**: Input char bigrams with highest activations for baseline model CHAR ([U]:URL, [N]:username, [R]:number, _:whitespace).

grams generate higher activation values but only for a few filters. Both types hold important clues in understanding our model. We use the intermediate representation of the CNN filters, consisting of a matrix $O \in \mathbb{R}^{n \times m}$ where n is the number of n-grams and m is the number of filters. We first determine the n-grams that generate the highest activation values aggregated over all filters. Table **3-6** shows the top 15 bigrams from this analysis for CNN-2 models trained on the whole dataset and on the refined dataset. Table **3-8** presents the top positive weighted features from the CHAR model. We can observe that many of the highest bigrams are uncommon versions of emoticons, such as *(:*, *:p* and *;D* that are likely correlated with specific authors. For the bot authors, *[U]* has the highest activation since most automated tweets have URLs at the end as their characteristic.

We then also collect the n-grams that have the highest number of filters where their activation is in the top 3. Table **3-7** shows the top bigrams from this analysis. Here we mostly see bigrams that are affixes. We can attribute this fact to the importance of morphological features for characterizing human tweets.

Authorship visualization using PCA

In order to visualize the workings of the CNN model, we pick out 4 authors, where two exhibit bot-like behavior and the other two exhibit human-like behavior. We use our CNN as feature extractor and propagate 100 tweets of each author through the network. This yields a vector of size 150 for each tweet, which we project to 2 dimensions using PCA. As can be seen in Figure 3.5, the CNN learns to clearly differentiate between two bot authors. We can also see that they follow very predictable patterns like: **" [URL]."** or **": [URL]"**, as discussed before. Similarly, in Figure 3.6, we are comparing two non-bot authors with a bot author. It is harder to distinctly separate the two non-bot authors, however the classifier is separating the bot and the non-bot instances very well.

3.4 Conclusions

We presented a strategy for using CNNs with character n-grams for AA of short texts, and provided a comprehensive comparison against standard approaches. We found that CNNs give better performance for AA of tweets, and using character n-grams instead of just character sequences can

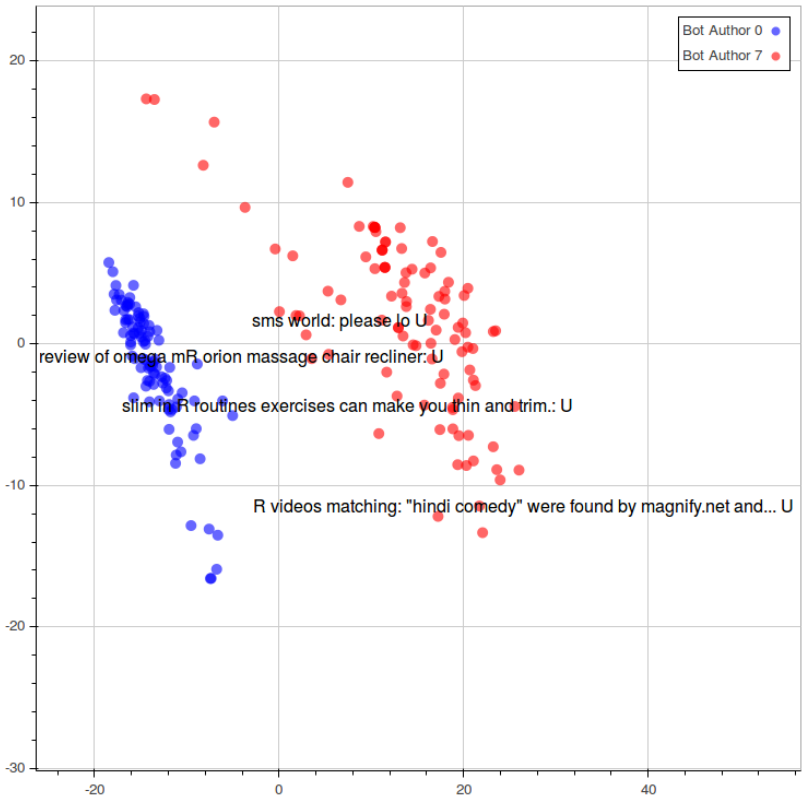

Figure 3.5: 2D Projection of tweets of two authors. Author 0 and Author 7 show a bot behavior. A clear separation is performed by the CNN.

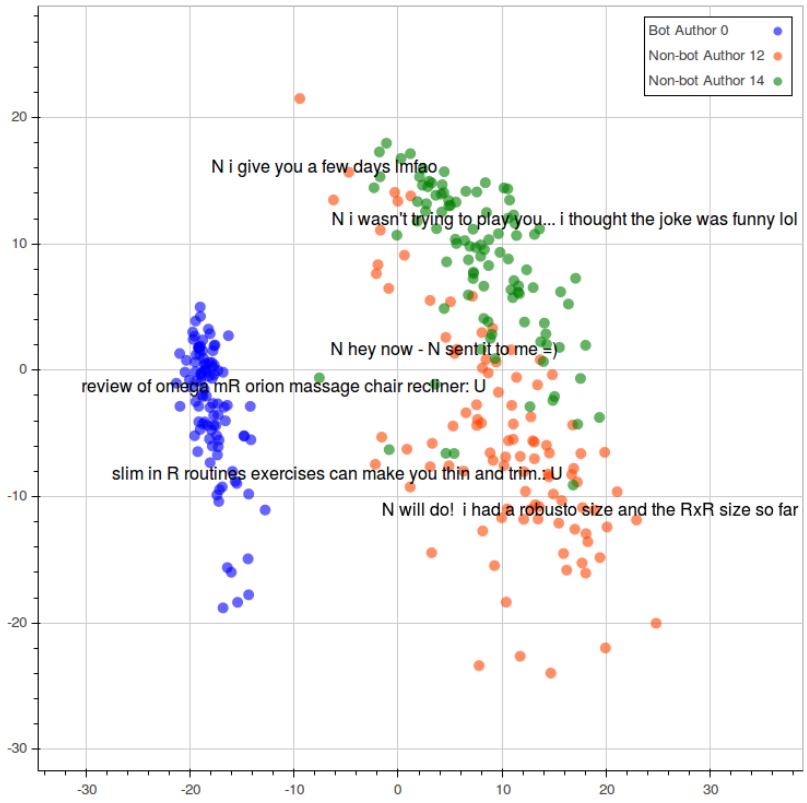

Figure 3.6: 2D Projection of tweets of three authors. Author 0 shows a bot behavior. Authors 12 and 14 show more human behavior

also improve performance. We were also able to gain some insights on what our architecture is actually learning. We could see that the network is focusing more on some sections of the text, making the model a perfect fit for attention models.

4 Using neural networks for author profiling

Author profiling task can vary from gender identification, age identification, personality traits identification and language variety identification. Each sub task is considered as a profile dimension of a social group. This chapter focuses on the description of the methods developed for solving the author profiling task. Author profiling task has been approached as an unimodal task, where the profile of an user is determined solely by his/her documents. Social media allows us to exploit not only text but images, video and audio. Thus, the profiling task has been recently approached as a multimodal task, where the profile is determined by the images and the text a user shares in social networks like Twitter or Facebook. For each problem, we propose a neural architecture for solving the profiling task. For unimodal author profiling, we propose a convolutional neural network, that uses sequences of words as input. This method was evaluated on the author profiling shared task of the laboratory on digital text forensics and stylometry (PAN). PAN 2017 author profiling shared task consisted on a corpus of Twitter users, where the objective was to determine the gender and the specific language variety of an user. Language variety is determined by the native country of an user. For the multimodal author profiling task, we adapted a gated based architecture for creating a rich multimodal representation of the texts and images of an user. Our adapted method was evaluated against information fusion approaches and unimodal approaches using an extended corpus of the PAN 2014 author profiling shared task. Part of this work was presented as part of the PAN shared tasks at the Conference and Labs of the Evaluation Forum (CLEF 2017):

1. **Sierra, S.**, Montes, M., Solorio, T., Gonzalez, F.A.. "Convolutional Neural Networks for Author Profiling". *In: Working Notes Papers of the CLEF* (2017)

4.1 Author profiling using short texts

The proposed method is similar to the **CNN-W** of the Section 3. The model uses sequences of words as input instead of character n-grams. Word inputs are capable of capturing content-based features rather than stylistic based features. In order to evaluate the performance of the model, a corpus of tweets from different languages was selected. The tweets were tokenized using a plain word tokenizer, while both case and stopwords were conserved during preprocessing. After that all the tweets of an author are concatenated and split into k evenly sized sequences of texts. Words are then represented by non-sparse vectors of dimension e, also known as embeddings. As Figure 4.1 shows, a sequence of words is represented as a matrix $C \in \mathbb{R}^{e \times k}$ where each column corresponds to the word embedding vector value.

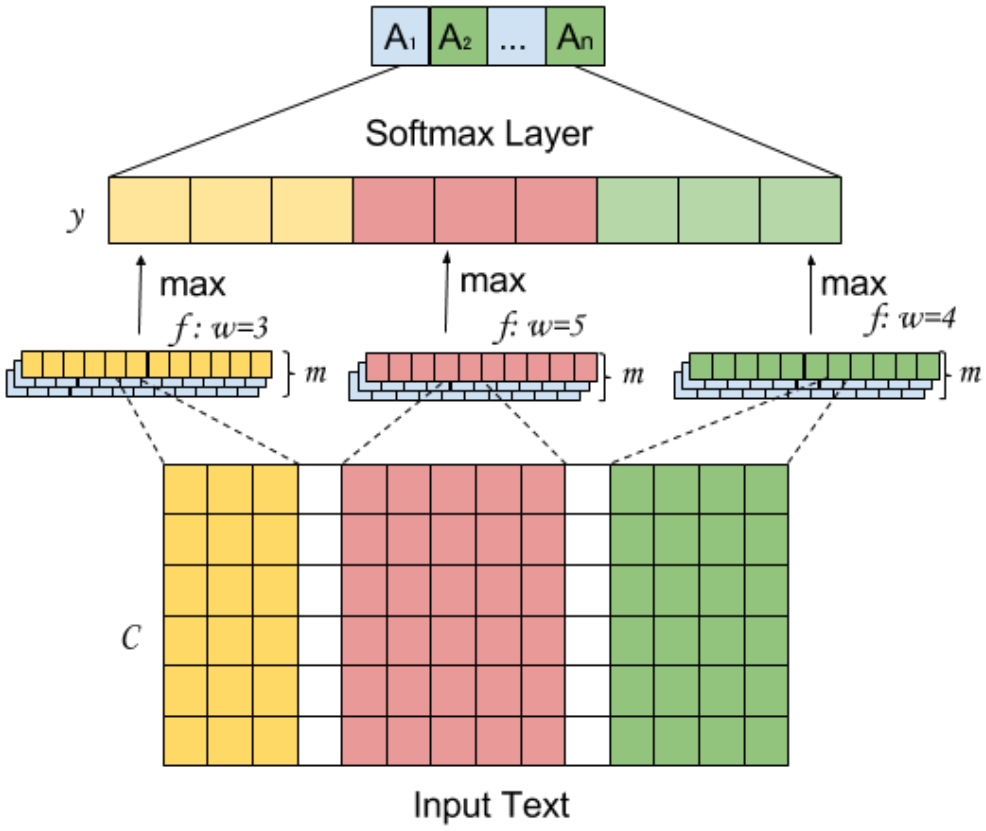

Figure 4.1: **CNN-W**. Word embeddings are fed to convolutional and max pooling layers, and the final classification is done via a softmax layer applied to the final text representation.

4.1.1 Convolutional Neural Networks using Words

Convolutional Neural Networks using words (CNN-W) receive a fixed-length sequence of words as input. Figure 4.1 depicts the CNN-W architecture. CNN-W first layer applies a set of convolutional filters of different sizes. For the concrete case of Figure 4.1 $m = \{500, 500, 500\}$ and $w = \{2, 3, 4\}$. The convolution operation performed by these filters is only applied in one dimension. Then a max-pooling over time operation is performed over the output feature maps, where only the maximum value of each feature map is used. The max pooling outputs for each feature map are concatenated in a vector. Figure 4.1 shows the output vector of size 1500 composed by the maximum activation values generated by each convolutional filter over the input. Finally, a softmax layer is added, where its size A_n depends on the profiling task. Dropout regularization was also used after the Embedding layer with a $p = 0.25$. Given that we train our network using sequences of text of one author, we used a bagging scheme for prediction stage. If we have n sequences of text for one author, we generate n predictions for the corresponding author, then we average the predictions and get the class with the highest value. In that way an author is labeled with its respective gender and language variety.

4.1.2 Experimental evaluation

Dataset

Our method was evaluated on the PAN AP 2017 shared task. This dataset consists of 10800 Twitter users. For each individual author, an XML document is provided along with his/her tweets. There are 3000 documents for English, 4200 for Spanish, 1200 for Portuguese and 2400 for Arabic. English tweets were written by native speakers from Australia, Canada, Great Britain, Ireland, New Zealand and United States. Spanish tweets were gathered from users from Argentina, Chile, Colombia, Mexico, Peru, Spain and Venezuela. Portuguese tweets did only come from Brazil and Portugal. Finally, Arabic tweets were collected from users that spoke four variants: Egypt, Gulf, Levantine and Maghrebi. Each author in the dataset has associated a gender (male or female) and language variety.

Experimental setup

For each language, we trained separately a model for gender and for language variety. For evaluation, we generated a stratified train/val split for every possible combination of **language_gender** and **language_variety**. Ten percent of the training documents was used for validation purposes. The evaluation of the models for the shared task was performed using TIRA [97]. TIRA allows both organizers and participants to have a common framework for evaluation. Also, participants of a shared task can deploy and evaluate their method without accessing directly to the test dataset. We deployed on TIRA the best model found by validation.

Results

Hyperparameters for **CNN-W** were explored in order to find the sub-optimal combination of parameters for solving the profiling task. CNN-W architecture was explored at two levels: Input-

Input Size	English		Spanish		Portuguese		Arabic	
	Gender	Variety	Gender	Variety	Gender	Variety	Gender	Variety
50	0.78	0.83	0.75	0.94	0.83	0.99	0.75	0.80
200	0.79	0.87	0.75	0.95	0.89	0.99	0.73	0.80
300	0.79	0.86	0.77	0.95	0.87	0.99	0.71	0.81

Table **4-1**: Validation results using different input sizes on **CNN-W**.

Filter sizes	English		Spanish		Portuguese		Arabic	
	Gender	Variety	Gender	Variety	Gender	Variety	Gender	Variety
$[2, 3]$	0.79	0.86	0.77	0.95	0.82	0.99	0.73	0.80
$[2, 3, 4]$	0.80	0.85	0.76	0.95	0.86	0.99	0.72	0.81

Table **4-2**: Validation results using different filter sizes on **CNN-W**.

level and convolution-level. For Input-related parameters we explored the type of input, the size of the input and the initialization values of the embeddings. The type of input was either tokenized sequences of words or sequences of character n-grams. The size of the input also was explored from a set of possible values $\{50, 200, 300\}$. Larger input sizes mean a reduction in the number of training samples, making the training process difficult for complex architectures. On Table **4-1**, we observed that 200 was a sub-optimal value for the input size. In some cases, 300 showed a competitive performance, but a larger input size also increases the number of parameters that the neural network has to learn. Initialization values of the embeddings were also evaluated using either pretrained embeddings or embeddings trained from scratch using the supervised signal of the profiling task. Pretrained word embeddings were trained on Wikipedia for every language using FastText [98]. As can be seen on Table **4-4**, pretrained embeddings improved the results.

For convolution-related parameters we explored the size w of the kernels and the number of kernels m. Larger size of kernels implies capturing long distance relationships between words, however this is only possible with a sufficient amount of training samples. Accordingly, we explored w from the set of values $\{2, 3\}, \{2, 3, 4\}$, while the number of filters m varied from 1500 up to 3000. As can be seen on Table **4-2**, different sets of values show competitive performance and both obtain high accuracy results on different language setups. On the other side, using a large number of filters, increases the representational capacity of the architecture, however it overfits quickly.

These architecture hyperparameters were found by exploration on the validation split of each setup and the best combination of parameters can be found in Table **4-3**. We found also that word-based inputs performed better than char-based inputs over all the profiling setups. For training, we employed Keras [99]. We shuffled the samples into mini-batches of size 32 and used Gradient Descent with Adaptive Moment Estimation [91] with default learning rate. Validation loss was monitored during 100 epochs and only models with the best validation accuracy were saved and used for testing.

4.1.3 Discussion

As can be seen on Table **4-4**, we compared CNN-W against other convolutional architectures. CNN-1 and CNN-2 use the same hyperparameters employed in the model for solving authorship

Layer	Parameters	English		Spanish		Portuguese		Arabic	
		Gender	Variety	Gender	Variety	Gender	Variety	Gender	Variety
Input	Input type	word	word	word	word	word	word	word	word
	Input size	200	200	300	200	200	200	50	300
	Pre-trained	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
Convolutional	m	1500	1500	1500	1500	1500	1500	1500	1500
	w	$[2,3,4]$	$[2,3]$	$[2,3]$	$[2,3,4]$	$[2,3,4]$	$[2,3,4]$	$[2,3]$	$[2,3,4]$

Table **4-3**: Best combination of hyperparameters for the neural network architecture. Possible values for the hyperparameters are as follows: Input type can be word or char. Input size varied from $\{50, 200, 300\}$. Pre-trained defined if embeddings were trained previously from FastText or not. Convolutional number of filters m varied from $\{1500, 3000\}$. Convolutional sizes of filters w comprised $\{2,3\}, \{2,3,4\}$

.

Method	English		Spanish		Portuguese		Arabic	
	Gender	Variety	Gender	Variety	Gender	Variety	Gender	Variety
BOW	**0.81**	0.79	0.73	0.89	0.85	**0.99**	0.73	0.78
CNN-W	0.79	0.87	0.75	0.95	**0.89**	**0.99**	0.73	0.80
CNN-W-FastText	0.80	**0.88**	**0.79**	**0.96**	0.86	**0.99**	**0.75**	**0.82**
CNN-1	0.75	0.73	0.66	0.84	0.80	**0.99**	0.68	0.79
CNN-2	0.73	0.77	0.71	0.89	0.85	**0.99**	0.73	0.81

Table **4-4**: Validation results using different architectures. **CNN-1** and **CNN-2** are character input based architectures. **CNN-W** and **CNN-W-FastText** are both trained on word inputs. **CNN-W-FastText** uses pretrained word embeddings for its respective language. **BOW** is a standard Bag-Of-Words

attribution in the Chapter 3. CNN-1 uses character unigrams as input for the convolutional neural network, while CNN-2 uses character bigrams as input. CNN-W was the best performing model as the result of the hyperparameter exploration on Tables **4-1** and **4-2**. The same architecture was trained using pretrained word embeddings. FastText [98] reported a method for creating rich word representations. We adopted previously trained models as the initial values of CNN-W and found that in most cases pretrained embeddings outperformed the previous results. In English, we found that BOW performed the best for the gender identification task with 81% of accuracy, however for language variety identification, CNN-W-FastText performs better with 88% of accuracy. For Spanish, CNN-W-FastText obtains the highest performance on both gender and language variety identification. CNN-W-FastText even outperforms BOW by a large margin, specially on language variety. Spanish language variety is the task with the highest number of classes, but we can argue that words shared by authors form the same country are very distinctive.

Portuguese variety consisted only of detecting tweets from Portugal and Brazil. Each method reached a 99% accuracy, thus, there are very specific words for each variety of Portuguese. For gender detection, pretrained embeddings did not outperform CNN-W, however it reached a competitive performance. In the Arabic setup, CNN-W-FastText got the best performance for both gender and language variety identification tasks. In the PAN shared task, we were able to evaluate our best performing method (CNN-W-FastText) on the test split. The test split was not released to the public, so our method was deployed on the TIRA evaluation system. Table **4-5** shows the

Language	Joint	Gender	Variety
English	0.66	0.78	0.84
Spanish	0.73	0.77	0.94
Portuguese	0.81	0.82	0.98
Arabic	0.57	0.68	0.79

Table **4-5**: Accuracy results on test dataset.

performance results of our method in the test dataset for the four languages. Accuracy is calculated separately for gender and language variation. For the joint column, accuracy is calculated on the basis that both gender and language variation were properly predicted.

Table **4-5** indicates how the Arabic setup was challenging for our best performing method

Our architecture was evaluated over sequences of words and characters. We found experimentally better validation performances using word sequences with pretrained word embeddings. Although we also found that training a CNN for author profiling produces additional challenges such as hyperparameter tuning and quick overfitting. In our parameter exploration we encountered models that were prone to overfit at the very first epochs. We solved this introducing dropout regularization or using an architecture with a fewer number of parameters. Also, evaluating our method on a challenging dataset which included different languages, shows how competitive our method is.

4.2 Multimodal author profiling

Text categorization applied to short texts imply a preprocessing step for extracting meaningful features. Moreover, Twitter restricts the number of characters a user can write on, forcing users to shorten some expressions and commit spelling mistakes. As can be seen on Figure 4.3, we defined a methodology for addressing the AP task. A tweet can be composed of merely text, an image next to a comment, a single image or a retweet. Figure 4.3 illustrates how text and images can appear on a single tweet. Our multimodal approach consists on finding the best representation for each modality and then both are combined using a gated multimodal unit (GMU) [100]. GMUs are capable of learning a rich multimodal representation. At the end of the section, we compare our adapted method against unimodal results and information fusion strategies.

4.2.1 Unimodal and multimodal representation methods

Here, we describe the features that were extracted from each modality. We also discuss three information fusion strategies.

Textual representations Common preprocessing of tweets includes filtering of URL's, accent removal, lowercasing and non-latin character removal. Furthermore, the AP task requires additional preprocessing steps. These involve hashtags and user mentions removal. While hashtags can be useful for predicting the topic an user is writing about, hashtags can also contain misleading information, *e.g.*, **#NotGuilty**, **#NOvsATL**. Below in Subsection 4.2.2, we describe the pipeline used for extracting the textual features.

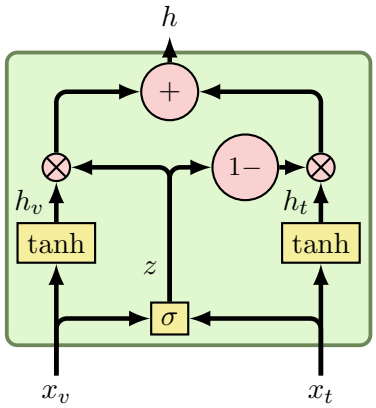

Figure 4.2: Gated multimodal unit. x_v and x_t vectors are the visual and textual representations respectively and h is the learned representation [100].

Visual representations Images in Twitter can be classified in three categories: profile pics, user-uploaded images and external images coming from re tweets. Profile pics are usually informative about the gender of the user [83], while [101] found that they are useful depending on the task (age or gender). Using features transfered from a deep CNN trained on another domain allows us to extract robust enough features. VGG-16 [102] is a deep CNN trained on natural images for the ILSVRC competition [103]. This network has been used as a feature extractor and has proved to learn highly discriminative features. VGG-16 is composed of two fully connected layers, prior its final softmax layer. Each of them is composed of 4096 neurons. Each image is scaled to 227×227 keeping its aspect ratio and fed to the VGG-16 network. Then, the activation values of the last fully connected layer are collected. A profile representation is built upon the average of the VGG-16 extracted features. A vector of 4096 dimensions will represent each user.

Fusion strategies As depicted in Figure 4.3, both text and images are extracted separately from the tweets, then, the best set of features for each modality is stored. Three strategies of fusion are tested:

- **Early fusion:** Using the best set of features for each modality, a representation is built for each user concatenating both modalities. L-2 Normalization is applied over the resulting vector. Then, a classifier is built on top of this representation.
- **Late fusion:** A classifier is trained on top of the best set of features for each modality. Then, predictions from both classifiers are combined using a soft scheme, where both predictions are averaged and the majority class will be chosen as final prediction.
- **Learned strategies:** A multimodal representation is learned using a Gated Multimodal Unit (GMU) [100], a neural network architecture that learns an intermediate representation combining both modalities. During the training process, GMUs are capable of adjusting their weights, by selecting which parts of the input contribute effectively to solve the AP task. Figure 4.2 depicts how the gates and the representations are computed.

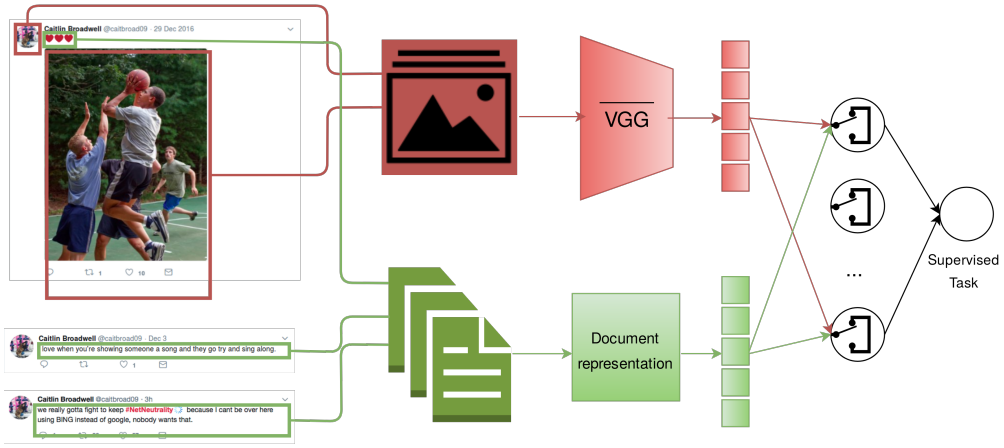

Figure 4.3: Different sources of information that are used for the multimodal author profiling task.

Language	# of users	# of images
Spanish	171	67539
English	280	75571

Table **4-6**: Distribution of users for the PAN 2014 AP extended corpus.

Classifiers Logistic Regression and Random Forest were used for the unimodal approaches. Also, both classifiers were tested for the late and early fusion strategies. Backpropagation was used for training the GMU architecture.

4.2.2 Experimental evaluation

Dataset

In this section, we describe the corpus used for multimodal author profiling. Álvarez-Carmona et al. [101] gathered an extended version of the PAN 2014 AP twitter corpus, including all the images that appeared on the users' profiles. Table **4-6** presents a general summary of the MAP corpus, including 279 user profiles in English and 171 in Spanish. Each user has associated a gender {male, female} and an age range {18-24, 25-34, 35-49, 50-64, 65-N}.

Experimental setup

Our approach addresses two profiling tasks: Gender and age identification. For each of them, we split the dataset in 70% and 30% training and test set respectively, keeping the proportion for each class by stratified sampling. From the training subset we took 20% as validation set to explore hyperparameters of the GMU.

Textual features

The PAN 2014 dataset contains information related to the tweets such as: type of tweet (retweet, comment, normal tweet), language and timestamp. However, inspecting manually some of the

tweets collected in the corpus, we found spanish native speakers in the set of english users. In order to evaluate the effect of the use of english or spanish, there were generated three overlapping sets of documents for each user:

- **All**: All the tweets.
- **Lang**: Only tweets labeled as written in english or spanish (Depending of the PAN AP language variation).
- **No-RT**: All the tweets, filtering out the retweets.

For each of these sets, we extracted also a different set of features described in Section 4.2.1. In order to compare with [101], we also generated a BoW representation, using only the most frequent unigrams. They are labeled as **Bow-2k** and **Bow-10k**, where 2k and 10k correspond to the considered number of unigrams. After building a textual representation for each user, these are fed to a classifier.

Visual features

In order to provide a fair comparison with [101], two sets were built using the user's images:

- **All**: Profile pics + user images + external images
- **Personal**: Profile pics + user images

Set *All* encompasses all the images from an user, while set *Personal* only contains those images considered as personal and *user-generated.*

Parameter selection

Stochastic gradient descent with ADAM optimization [104] was used to learn the weights of the neural network. Dropout and max-norm regularization were used to control overfitting. Hidden size ($\{64, 128, 256, 512\}$), learning rate ($\left[10^{-3}, 10^{-1}\right]$), dropout ($[0.3, 0.7]$), max-norm ($[5, 20]$) and initialization ranges ($\left[10^{-3}, 10^{-1}\right]$) parameters were explored by training 25 models with random (uniform) hyperparameter initializations and the best was chosen according to validation performance. It has been reported that this strategy is preferable over grid search when training deep models [105]. All the implementation was carried on with the Tensorflow framework.

Results

Unimodal results

As can be seen on Table **4-9**, there are three groups of textual features. **BOW-baseline** is built using the same setup described by [106]. Modaresi, Liebeck, and Conrad [106] extract a series of stylistic and content based features. Content features are extracted by a Bag of word bigrams and unigrams. Stylistic features are extracted using a bag of character n-grams. Since [106] applies all these features at once for author profiling, we evaluated the effect of each feature separately. Therefore, content features are extracted in **BOW-unigrams** and **BOW-bigrams**.

Table **4-7**: Results for gender task using only the text modality in the English set of users for PAN AP corpus

Text Features	**All-text**	**no-rt**	**only-lang**
BOW-baseline	0.750	0.785	0.761
BOW-unigrams	0.785	0.785	0.845
BOW-bigrams	0.690	0.700	0.690
BOW-char	0.738	0.738	0.770
BOW-unigram_2k	0.869	**0.892**	**0.892**
BOW-unigram_10k	0.845	0.830	0.845
BOW-bigram_2k	0.773	0.750	0.821
BOW-bigram_10k	0.809	0.797	0.830

Table **4-8**: Results for gender task using only the text modality in the Spanish set of users for PAN AP corpus

Text Features	**All-text**	**no-rt**	**only-lang**
BOW-baseline	0.765	0.804	0.569
BOW-unigrams	0.745	**0.843**	0.647
BOW-bigrams	0.706	0.725	0.608
BOW-char	0.765	0.784	0.549
BOW-unigram_2k	0.824	**0.843**	0.588
BOW-unigram_10k	0.804	**0.843**	0.627
BOW-bigram_2k	0.804	0.784	0.569
BOW-bigram_10k	0.765	0.824	0.569

Stylistic features are extracted using the **BOW-char**. Additionally, we compare to [101], using the same bag representation of unigrams, but using only top 2.000 (2k) and 10.000 (10k) most frequent words. As can be seen Table **4-7** and **4-8**, the best representation for the gender task is **BOW-unigram_2k**. Additionally, considering only the tweets generated by the user, that means, without including retweets, generates better results. In Table **4-9** and **4-10**, the concatenation of content and stylistic features (**BOW-baseline**) generates the best results for the age task. However, considering only tweets in English improves the accuracy of the age classification method. For Spanish, considering tweets generated by the user produces the age task accuracy.

Visual modality results are very poor compared to the textual modality results. Table **4-11** shows the performance of VGG extracted features in the English split. The results show that using all the images is better for the age and gender classification task. While Table **4-12** shows that the choice of source images is only relevant for solving the age classification task.

Multimodal results

Standard multimodal strategies involve early and late fusion strategies. Table **4-13** shows the summary of results for early, late and GMU methods for English. GMU outperformed the other

Table **4-9**: Results for age task using only the text modality in the English set of users for PAN AP corpus

Text Features	**All-text**	**no-rt**	**only-lang**
BOW-baseline	0.488	0.511	**0.5357**
BOW-unigrams	0.511	0.488	0.500
BOW-bigrams	0.488	0.488	0.464
BOW-char	0.500	0.500	0.511
BOW-unigram_2k	0.476	0.452	0.464
BOW-unigram_10k	0.488	0.452	0.488
BOW-bigram_2k	0.428	0.404	0.464
BOW-bigram_10k	0.488	0.488	0.488

Table **4-10**: Results for age task using only the text modality in the Spanish set of users for PAN AP corpus

Text Features	**All-text**	**no-rt**	**only-lang**
BOW-baseline	0.490	**0.529**	0.490
BOW-unigrams	0.471	0.490	0.490
BOW-bigrams	0.490	0.490	0.490
BOW-char	0.490	0.510	0.510
BOW-unigram_2k	0.373	0.412	0.392
BOW-unigram_10k	0.471	0.431	0.451
BOW-bigram_2k	0.392	0.471	0.451
BOW-bigram_10k	0.471	0.471	0.510

Table **4-11**: Results for age and gender task using visual strategies in the English set of users for PAN AP corpus

Image Features strategy	Gender	Age
VGG-AVG	**0.796**	**0.428**
VGG-AVG-Personal	0.761	0.345

Table **4-12**: Results for age and gender task using visual strategies in the Spanish set of users for PAN AP corpus

Image Features strategy	Gender	Age
VGG-AVG	**0.647**	0.314
VGG-AVG-Personal	**0.647**	**0.373**

Table **4-13**: Results for age and gender task using multimodal strategies in the English set of users for PAN AP corpus

Multimodal strategy	**Gender**	**Age**
GMU	**0.892**	**0.571**
Late Fusion	0.869	0.500
Early Fusion	0.821	0.381

Table **4-14**: Results for age and gender task using multimodal strategies in the Spanish set of users for PAN AP corpus

Multimodal strategy	**Gender**	**Age**
GMU	**0.863**	**0.569**
Late Fusion	0.804	0.353
Early Fusion	0.667	0.392

strategies in both tasks: Gender and Age identification. The difference is bigger for Age identification task. This is a more complex task because it is an imbalanced multiclass problem. The GMU was able to take advantage of both modalities better than standard fusion strategies. In the Spanish setup, GMU also obtained a better performance than standard fusion approaches. As can be seen on Table **4-14**, early and late fusion strategies in the age task are not very competitive. In Table **4-15**, the summary of the results per each modality and for the multimodal approaches is presented. GMU also is capable of obtaining a better performance than unimodal approaches.

Table **4-15**: Summary for the gender and age task in the multimodal PAN AP 2014 corpus

		English		Spanish	
Modality	**Representation**	**Gender**	**Age**	**Gender**	**Age**
Multimodal	GMU	**0.892**	**0.571**	**0.863**	**0.569**
	Late Fusion	0.869	0.500	0.804	0.353
	Early Fusion	0.821	0.416	0.667	0.392
Textual	BOW-unigrams_2k_no_rt	0.892	0.452	0.843	0.412
	BOW-unigrams_no_rt	0.785	0.488	0.843	0.490
	BOW-baseline_only_lang	0.761	0.536	0.569	0.490
	BOW-baseline_no_rt	0.785	0.511	0.804	0.529
Visual	VGG-AVG	0.791	0.428	0.647	0.314
	VGG-AVG-personal	0.761	0.345	0.647	0.373

4.2.3 Discussion

We proposed a new pipeline for addressing profiling tasks on twitter social network. *Age* and *Gender* properties are predicted using visual and textual information. The VGG network was used to get the visual representation from a set of tweeted images, while different text-based representations were

Table **4-16**: Accuracy per class for the age task in the multimodal PAN AP 2014 corpus

Modality	**Representation**	**18-24**	**25-34**	**35-49**	**50-64**	**>=65**
Multimodal	Early Fusion	0.000	0.458	0.513	0.125	0.000
	Late Fusion	0.000	0.625	0.729	0.000	0.000
	GMU	0.000	**0.750**	**0.757**	**0.188**	0.000
Textual	BOW-baseline_only_en	0.000	**0.750**	0.730	0.000	0.000
	BOW-unigrams_2k_only_en	0.000	0.583	0.648	0.062	0.000
	BOW-unigrams_only_en	0.000	0.667	0.703	0.000	0.000
Visual	VGG-AVG	0.000	0.542	0.567	0.125	0.000
	VGG-AVG-personal	**0.200**	0.417	0.459	0.062	0.000

adapted for the particularities of textual content in the tweets. To combine both representation, we also explored different standard fusion strategies along with the GMU network. This model has proven to learn good intermediate representations for supervised tasks that involve multiple information sources.

Experimental results showed unigrams of words perform better than other n-grams. Also, the combination of visual and textual representation outperforms unimodal approaches. Table **4-16** shows one common problem of all the representation strategies for the age task. This problem consisted of the low accuracy value per class for imbalanced classes. Although GMU obtains the best performance overall it still fails at classifying users from less frequent classes. We also observed that visual representations performed very poorly compared to textual representations. Finally, the GMU network improves average accuracy from 0.86 to 0.89 in gender identification for English and from 0.50 to 0.57 in age classification. For Spanish, the GMU network improved average accuracy from 0.80 to 0.86 in gender identification and from 0.39 to 0.56 in age classification.

4.3 Conclusions

Although, author profiling tasks usually consists of a large number of documents, at prediction time it only comes to determine the profile of the authors of the documents. One author can have as many documents as possible, then, a prediction for each document can be made, and all the predictions have to be combined to generate a single prediction. We used convolutional neural networks for extracting rich representations, but our models were easy to overfit and strong regularization strategies were employed like dropout. Before prediction time, we gathered all the predictions for the documents of one author and generated a single prediction. This combination sometimes was not optimal and remains as an unexplored field. In spite of the exposed problems, we were able to propose a competitive neural network for approaching the text based author profiling task.

Bagging of predictions was a similar issue in the multimodal approach. For instance, in the English setup of the PAN AP 2014 corpus, after generating one representation for each author, the number of samples for training was only 196. Therefore our model quickly overfitted, however, we

found a configuration using dropout where our model was able to outperform standard information fusion strategies. Next steps in this research involve exploring end-to-end models, where we can exploit the true nature of each modality, instead of using a compressed representation of each one.

5 Conclusions and future work

5.1 Conclusions

In this thesis, different neural network architectures were applied for addressing two different authorship analysis tasks: author profiling and authorship attribution. For authorship attribution, we proposed a method for detecting the authorship of a list of short texts in the domain of Twitter. Our approach was evaluated in terms of variability of the number of candidate authors and also was evaluated in terms of the number of available training instances for each author. Our method consisted of a convolutional neural network architecture that exploited character n-grams based inputs. Our architecture was compared also against standard approaches for representing texts in authorship attribution domain and different convolutional-based architectures. Our proposed architecture achieved the best performance against the other methods in every proposed setup. For author profiling, we explored two different setups: unimodal profiling using only text and multimodal author profiling using text and images. For unimodal profiling, we proposed an architecture which was similar to the proposed for addressing the authorship attribution task. A convolutional neural network was proposed using sequences of words as input. Sequences of words are more suitable for capturing content-based features. Our approach was compared against standard Bag-of-Words approach and other different convolutional based architectures. Our best performing method was submitted to the author profiling shared task of the 2017 edition of the laboratory on digital text forensics and stylometry (PAN). The dataset for evaluating unimodal author profiling consisted of a series of Twitter authors coming from different countries. The main objective was to predict both the gender of the author and the specific variation of language. Variations of language are determined by the native country of the Twitter user. For multimodal author profiling, we employed an extended dataset of the author profiling shared task of the 2014 edition of the PAN laboratory. For this corpus, a huge amount o.

Using learned representations through neural approaches obtained better results in authorship analysis. Feature extraction in authorship analysis is a key step, and we were able to propose methods for learning a representation while we solved the supervised task: either profiling or authorship attribution. Next we detail the main conclusions of this work:

1. In authorship attribution, we presented a neural network that outperformed standard approaches in several setups where the number of training instances varied, as well as the number of authors (classes) varied.

2. Authorship attribution techniques take a huge advantage of the type of input representation. In this thesis, we explored word based and character based inputs and found that character-based inputs provide a better performance when approaching AA tasks.

3. Saliency models are able to provide valuable insights in neural networks based approaches. In the AA section, we compared what type of character n-grams affected the performance of the proposed convolutional neural network. Interpretability is indeed one of the most important factors for using a method to solve a task.

4. Unlike authorship attribution, author profiling techniques work better using word-based inputs. This fact can be due to the content words that different demographic groups. Gender based groups tend to use different words. People from different groups tend to use also very specific words.

5. We showed also that using previously learned embeddings improved the performance of our CNN in the unimodal author profiling task.

6. For multimodal author profiling, we found that GMUs are capable of building a multimodal representation that is able of outperforming standard information fusion approaches: early and late fusion.

5.2 Future Work

There are still issues regarding to the application of representation learning techniques to authorship analysis tasks. There exists a plenty of neural networks based models, however most of them have a huge amount of parameters to be estimated, which makes it an easy-to-overfit task. As future work, we propose possible directions of future research:

1. Evaluating attention based models in order to address authorship analysis tasks. Attention based models have the great advantage of learning automatically which parts of the input are more important for the supervised task.

2. For the authorship attribution, we only evaluated a single genre attribution scenario, however there is an increasing trend in the research community towards cross genre authorship attribution. It is important to isolate those style patterns that are inherent to the author's writing style and not dependent to the document's genre.

3. For multimodal author profiling, we were able to improve results over single modality profiling, however visual representation was not properly used for characterizing age or gender. This is difficult because there is not a standard way of representing a *visual document* for an author, i.e., to find a proper way for combining the images of an author and extract a combined representation.

4. For gated multimodal approaches, an end-to-end approach would be an alternative for the lack of samples at training time. In and end-to-end approach, a proper representation for text and image would be learned, while a rich multimodal representation is learned using the GMUs. One single instance for training would be a batch of tweets and images instead of a single vector combining all the documents and the images from one author.

Bibliography

[1] Joseph Turian, Lev Ratinov, and Yoshua Bengio. "Word representations: a simple and general method for semi-supervised learning". In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 384–394. URL: http://dl.acm.org/citation.cfm?id=1858721.

[2] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *Nature* 521.7553 (May 2015), pp. 436–444. DOI: 10.1038/nature14539. URL: http://dx.doi.org/10.1038/nature14539.

[3] John Rupert Firth. *A Synopsis of Linguistic Theory, 1930-1955*. 1957. URL: https://books.google.com.co/books?id=T8LDtgAACAAJ.

[4] Yoshua Bengio, Réjean Ducharme, and Vincent. "A Neural Probabilistic Language Model". In: *Journal of Machine Learning Research* 3.3 (2003), pp. 1137–1155. URL: http://link.springer.com/chapter/10.1007/3-540-33486-6%7B%5C_%7D6.

[5] Marco Baroni and Alessandro Lenci. "Distributional Memory: A General Framework for Corpus-based Semantics". In: *Computational Linguistics* 36.4 (2010), pp. 673–721. ISSN: 0891-2017. DOI: 10.1162/coli_a_00016. URL: http://dx.doi.org/10.1162/coli%7B%5C_%7Da%7B%5C_%7D00016.

[6] Yoshua Bengio et al. "Neural probabilistic language models". In: *Innovations in Machine Learning* (2006), pp. 137–186. ISSN: 14349922. DOI: 10.1007/10985687_6.

[7] Geoffrey E Hinton. "Learning distributed representations of concepts". In: *Proceedings of the eighth annual conference of the cognitive science society*. Vol. 1. Amherst, MA. 1986, p. 12.

[8] Tomas Mikolov et al. "Recurrent Neural Network based Language Model". In: *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association*. Makuhari, Chiba, Japan, 2010, pp. 1045–1048.

[9] Tomas Mikolov et al. "Extensions of recurrent neural network language model". In: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 5528–5531. URL: http://ieeexplore.ieee.org/xpls/abs%7B%5C_%7Dall.jsp?arnumber=5947611.

[10] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. "Learning Long-Term Dependencies with Gradient Descent is Difficult". In: *IEEE Transactions on Neural Networks* 5.2 (1994), pp. 157–166. ISSN: 1045-9227. DOI: 10.1109/72.279181.

[11] Frederic Morin and Yoshua Bengio. "Hierarchical probabilistic neural network language model". In: *Proceedings of the international workshop on artificial intelligence and statistics*. 2005, pp. 246–252. URL: http://core.kmi.open.ac.uk/download/pdf/22017.pdf%7B%5C#%7Dpage=255.

[12] Tomas Mikolov et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013). URL: http://arxiv.org/abs/1301.3781.

[13] Andriy Mnih and Yee Whye Teh. "A fast and simple algorithm for training neural probabilistic language models". In: *Proceedings of the 29th International Conference on Machine Learning*. 2012. URL: http://arxiv.org/abs/1206.6426.

[14] Ronan Collobert and Jason Weston. "A unified architecture for natural language processing: Deep neural networks with multitask learning". In: *Proceedings of the 25th international conference on Machine learning*. ACM. ACM, 2008, pp. 160–167. URL: http://dl.acm.org/citation.cfm?id=1390177.

[15] Jason Weston et al. "Natural Language Processing (Almost) from Scratch". In: *Journal of Machine Learning Research* 12 (2011), pp. 2461–2505. URL: http://dl.acm.org/citation.cfm?id=2078186%20http://infoscience.epfl.ch/record/192375/files/Collobert%7B%5C_%7DJMLR%7B%5C_%7D2493--2537.pdf.

[16] Tomas Mikolov et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in Neural Information Processing Systems*. 2013, pp. 3111–3119. arXiv: 1310.4546. URL: http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.

[17] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. "Linguistic regularities in continuous space word representations". In: *HLT-NAACL*. 2013, pp. 746–751. URL: https://www.aclweb.org/anthology/N/N13/N13-1090.pdf.

[18] Omer Levy and Yoav Goldberg. "Linguistic Regularities in Sparse and Explicit Word Representations". In: *Proceedings of the 18th Conference on Computational Natural Language Learning (CoNLL 2014)*. 2014, p. 171. ISBN: 9781937284473.

[19] Jeff Mitchell and Mirella Lapata. "Composition in distributional models of semantics". In: *Cognitive science* 34.8 (2010), pp. 1388–1429.

[20] William Blacoe and Mirella Lapata. "A comparison of vector-based representations for semantic composition". In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012, pp. 546–556. URL: http://dl.acm.org/citation.cfm?id=2391011.

[21] Marco Baroni and Roberto Zamparelli. "Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space". In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. 2010, pp. 1183–1193.

[22] Quoc Le and Tomas Mikolov. "Distributed Representations of Sentences and Documents". In: *Proceedings of the 31st International Conference on Machine Learning*. Vol. 32. 2014, pp. 1188–1196. ISBN: 9781634393973. arXiv: 1405.4053. URL: http://arxiv.org/abs/1405.4053.

[23] Jiwei Li, Minh-thang Luong, and Dan Jurafsky. "A Hierarchical Neural Autoencoder for Paragraphs and Documents". In: *Acl*. 2015. arXiv: arXiv:1506.01057v2.

[24] Ryan Kiros et al. "Skip-thought vectors". In: *Advances in Neural Information Processing Systems* (2015), pp. 3276–3284.

[25] Bill Dolan, Chris Quirk, and Chris Brockett. "Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources". In: *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics. 2004. DOI: 10.3115/1220355.1220406. URL: http://dl.acm.org/citation.cfm?id=1220406.

[26] Nitin Madnani, Joel Tetreault, and Martin Chodorow. "Re-examining machine translation metrics for paraphrase identification". In: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics. 2012, pp. 182–190.

[27] Wei Xu et al. "Extracting Lexically Divergent Paraphrases from Twitter". In: *Transactions of the Association for Computational Linguistics* 2 (2014), pp. 435–448.

[28] Alberto Barrón-Cedeño et al. "Plagiarism Meets Paraphrasing: Insights for the Next Generation in Automatic Plagiarism Detection". In: *Computational Linguistics* 39.4 (Jan. 2013), pp. 917–947. ISSN: 0891-2017. DOI: 10.1162/COLI_a_00153. URL: http://dx.doi.org/10.1162/COLI%7B%5C_%7Da%7B%5C_%7D00153.

[29] M Marelli et al. "A SICK cure for the evaluation of compositional distributional semantic models". In: *Proceedings of LREC 2014*. Ed. by ELRA. Reijjavik, 2014, pp. 216–223.

[30] Kai Sheng Tai, Richard Socher, and Christopher D Manning. "Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing.* (2015), pp. 1556–1566. arXiv: arXiv:1503.00075v3.

[31] Thomas Corwin Mendenhall. "The characteristic curves of composition". In: *Science* 9.214 (1887), pp. 237–249.

[32] Efstathios Stamatatos. "A survey of modern authorship attribution methods". In: *Journal of the American Society for Information Science and Technology* 60.3 (Mar. 2009), pp. 538–556. ISSN: 15322882. DOI: 10.1002/asi.21001. URL: http://doi.wiley.com/10.1002/asi.21001.

[33] Moshe Koppel, Jonathan Schler, and Shlomo Argamon. "Computational methods in authorship attribution". In: *Journal of the American Society for Information Science and Technology* 60.1 (Jan. 2009), pp. 9–26. ISSN: 15322882. DOI: 10.1002/asi.20961. URL: http://doi.wiley.com/10.1002/asi.20961.

[34] Shlomo Argamon et al. "Stylistic text classification using functional lexical features". In: *Journal of the American Society for Information Science and Technology* 58.6 (Apr. 2007), pp. 802–822. ISSN: 15322882. DOI: 10.1002/asi.20553. URL: http://doi.wiley.com/10.1002/asi.20553.

[35] Efstathios Stamatatos et al. "Overview of the PAN/CLEF 2015 evaluation lab". In: *International Conference of the Cross-Language Evaluation Forum for European Languages* (2015), pp. 518–538.

[36] Roy Schwartz et al. "Authorship attribution of micro-messages". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 2013, pp. 1880–1891.

[37] Robert Layton, Paul Watters, and Richard Dazeley. "Authorship Attribution for Twitter in 140 Characters or Less". In: *2010 Second Cybercrime and Trustworthy Computing Workshop*. IEEE, July 2010, pp. 1–8. ISBN: 978-1-4244-8054-8. DOI: 10.1109/CTC.2010.17. URL: http://ieeexplore.ieee.org/document/5615152/.

[38] Moshe Koppel and Yaron Winter. "Determining if two documents are written by the same author". In: *Journal of the Association for Information Science and Technology* 65.1 (Jan. 2014), pp. 178–187. ISSN: 23301635. DOI: 10.1002/asi.22954. URL: http://doi.wiley.com/10.1002/asi.22954.

[39] Farkhund Iqbal et al. "A novel approach of mining write-prints for authorship attribution in e-mail forensics". In: *Digital Investigation* 5 (2008), S42–S51. ISSN: 1742-2876. DOI: https://doi.org/10.1016/j.diin.2008.05.001. URL: http://www.sciencedirect.com/science/article/pii/S1742287608000315.

[40] Yanir Seroussi, Ingrid Zukerman, and Fabian Bohnert. "Authorship Attribution with Topic Models". In: *Computational Linguistics* 40.2 (2014), pp. 269–310. DOI: 10.1162/COLI_a_00173. eprint: https://doi.org/10.1162/COLI_a_00173. URL: https://doi.org/10.1162/COLI_a_00173.

[41] Michal Rosen-Zvi et al. "The Author-topic Model for Authors and Documents". In: *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*. UAI '04 (2004), pp. 487–494. URL: http://dl.acm.org/citation.cfm?id=1036843.1036902.

[42] Jacques Savoy. "Authorship attribution based on a probabilistic topic model". In: *Information Processing & Management* 49.1 (2013), pp. 341–354. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2012.06.003. URL: http://www.sciencedirect.com/science/article/pii/S0306457312000751.

[43] Upendra Sapkota et al. "Not All Character N-grams Are Created Equal: A Study in Authorship Attribution". In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (May 2015), pp. 93–102. URL: http://www.aclweb.org/anthology/N15-1010.

[44] Maciej Eder. "Does size matter? Authorship attribution, small samples, big problem". In: *Digital Scholarship in the Humanities* 30.2 (2015), pp. 167–182. ISSN: 2055-7671. DOI: 10.1093/llc/fqt066. eprint: http://dsh.oxfordjournals.org/content/30/2/167.full.pdf. URL: http://dsh.oxfordjournals.org/content/30/2/167.

[45] Efstathios Stamatatos. "Author identification: Using text sampling to handle the class imbalance problem". In: *Information Processing & Management* 44.2 (2008). Evaluating Exploratory Search Systems Digital Libraries in the Context of Users' Broader Activities, pp. 790–799. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2007.05.012. URL: http://www.sciencedirect.com/science/article/pii/S0306457307001197.

[46] Georgia Frantzeskou et al. "Identifying authorship by byte-level n-grams: The source code author profile (SCAP) method". In: *International Journal of Digital Evidence* 6.1 (2007), pp. 1–18.

[47] Tie-Yun Qian et al. "Review Authorship Attribution in a Similarity Space". In: *Journal of Computer Science and Technology* 30.1 (Jan. 2015), pp. 200–213. ISSN: 1000-9000. DOI: 10.1007/s11390-015-1513-6. URL: http://link.springer.com/10.1007/s11390-015-1513-6.

[48] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. "A Convolutional Neural Network for Modelling Sentences". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (June 2014), pp. 655–665. URL: http://www.aclweb.org/anthology/P14-1062.

[49] Yoon Kim. "Convolutional Neural Networks for Sentence Classification". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Oct. 2014), pp. 1746–1751. URL: http://www.aclweb.org/anthology/D14-1181.

[50] Xiang Zhang, Junbo Zhao, and Yann LeCun. "Character-level convolutional networks for text classification". In: *Advances in Neural Information Processing Systems* (2015), pp. 649–657.

[51] Ronan Collobert et al. "Natural language processing (almost) from scratch". In: *Journal of Machine Learning Research* 12.Aug (2011), pp. 2493–2537.

[52] Yoon Kim et al. "Character-Aware Neural Language Models". In: *AAAI Conference on Artificial Intelligence* (2016). URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12489.

[53] Dylan Rhodes. *Author Attribution with CNN's*. https://cs224d.stanford.edu/reports/RhodesDylan.pdf. 2015.

[54] Patrick Juola. "An Overview of the Traditional Authorship Attribution Subtask". In: *Proceedings of PAN/CLEF 2012* (2012).

[55] Yunita Sari, Andreas Vlachos, and Mark Stevenson. "Continuous N-gram Representations for Authorship Attribution". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers* (2017), pp. 267–273. URL: http://aclanthology.coli.uni-saarland.de/pdf/E/E17/E17-2043.pdf.

[56] Armand Joulin et al. "Bag of Tricks for Efficient Text Classification". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers* (2017), pp. 427–431. URL: http://aclanthology.coli.uni-saarland.de/pdf/E/E17/E17-2068.pdf.

[57] Wenpeng Yin and Hinrich Schütze. "Multichannel Variable-Size Convolution for Sentence Classification". In: *Proceedings of the Nineteenth Conference on Computational Natural Language Learning* (July 2015), pp. 204–214. URL: http://www.aclweb.org/anthology/K15-1021.

[58] Sebastian Ruder, Parsa Ghaffari, and John G Breslin. "Character-level and Multi-channel Convolutional Neural Networks for Large-scale Authorship Attribution". In: *arXiv preprint arXiv:1609.06686* (2016).

[59] Shlomo Argamon et al. "Automatically Profiling the Author of an Anonymous Text". In: *Commun. ACM* 52.2 (2009), pp. 119–123. ISSN: 0001-0782. DOI: 10.1145/1461928.1461959. URL: http://doi.acm.org/10.1145/1461928.1461959.

[60] JK Chambers and N Schilling. *The handbook of language variation and change*. Blackwell Publishing, 2013. URL: https://books.google.com.co/books?hl=en%7B%5C&%7Dlr=%7B%5C&%7Did=HJLeCRds-6AC%7B%5C&%7Doi=fnd%7B%5C&%7Dpg=PT2%7B%5C&%7Ddq=he+Handbook+of+Language+Variation+and+Change.%7B%5C&%7Dots=fw0PFwYz8E%7B%5C&%7Dsig=ytL8KIRR0xGHtCtsBu%7B%5C_%7Dhav5C2rI.

[61] Jonathan Schler et al. "Effects of age and gender on blogging." In: *AAAI spring symposium: Computational approaches to analyzing weblogs*. 2006, pp. 199–205. URL: http://www.aaai.org/Papers/Symposia/Spring/2006/SS-06-03/SS06-03-039.pdf.

[62] Shlomo Argamon et al. "Gender, Genre, and Writing Style in Formal Written Texts". In: *Text - Interdisciplinary Journal for the Study of Discourse* 23.3 (2003). DOI: 10.1515/text.2003.014.

[63] Francisco Manuel Rangel Pardo et al. "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*. Ed. by Linda Cappellato et al. Vol. 1866. CEUR Workshop Proceedings. CLEF and CEUR-WS.org, 2017. URL: http://ceur-ws.org/Vol-1866/.

[64] Martin Potthast et al. "Overview of PAN'17: Author Identification, Author Profiling, and Author Obfuscation". In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction. 7th International Conference of the CLEF Initiative (CLEF 17)*. Ed. by Gareth J. F. Jones et al. Berlin Heidelberg New York: Springer, Sept. 2017.

[65] Moshe Koppel, Shlomo Argamon, and Anat Rachel Shimoni. "Automatically Categorizing Written Texts by Author Gender". In: *Literary and Linguistic Computing* 17.4 (Nov. 2002), pp. 401–412. ISSN: 0268-1145. URL: http://dx.doi.org/10.1093/llc/17.4.401.

[66] Shlomo Argamon et al. "Automatically Profiling the Author of an Anonymous Text". In: *Communications of the ACM* 52.2 (2009), pp. 119–123. ISSN: 0001-0782. DOI: 10.1145/1461928.1461959. URL: http://doi.acm.org/10.1145/1461928.1461959.

[67] John D. Burguer et al. "Discriminating gender on Twitter". In: *Proceedings of the conference on empirical methods in natural language processing*. 2011. URL: https://dl.acm.org/citation.cfm?id=2145568.

[68] Claudia Peersman, Walter Daelemans, and Leona Van Vaerenbergh. "Predicting Age and Gender in Online Social Networks". In: *Proceedings of the 3rd International Workshop on Search and Mining User-generated Contents*. SMUC '11. New York, NY, USA: ACM, 2011, pp. 37–44. ISBN: 978-1-4503-0949-3. DOI: 10.1145/2065023.2065035. URL: https://dl.acm.org/citation.cfm?id=2065035%20http://doi.acm.org/10.1145/2065023.2065035.

[69] A. Pastor López-Monroy et al. "Discriminative subprofile-specific representations for author profiling in social media". In: *Knowledge-Based Systems* 89 (Nov. 2015), pp. 134–147. ISSN: 0950-7051. DOI: 10.1016/J.KNOSYS.2015.06.024. URL: http://www.sciencedirect.com/science/article/pii/S0950705115002427.

[70] Rosa María Ortega-Mendoza et al. "Emphasizing personal information for Author Profiling: New approaches for term selection and weighting". In: *Knowledge-Based Systems* 145 (Apr. 2018), pp. 169–181. DOI: 10.1016/J.KNOSYS.2018.01.014. URL: https://www.sciencedirect.com/science/article/pii/S0950705118300224.

[71] Francisco Rangel and Paolo Rosso. "On the impact of emotions on author profiling". In: *Information Processing & Management* 52.1 (Jan. 2016), pp. 73–92. DOI: 10.1016/J.IPM.2015.06.003. URL: http://www.sciencedirect.com/science/article/pii/S0306457315000783.

[72] A. Pastor López-Monroy et al. "Discriminative subprofile-specific representations for author profiling in social media". In: *Knowledge-Based Systems* 89 (Nov. 2015), pp. 134–147. ISSN: 0950-7051. DOI: 10.1016/J.KNOSYS.2015.06.024. URL: http://www.sciencedirect.com/science/article/pii/S0950705115002427.

[73] Rosa María Ortega-Mendoza et al. "Emphasizing personal information for Author Profiling: New approaches for term selection and weighting". In: *Knowledge-Based Systems* 145 (Apr. 2018), pp. 169–181. DOI: 10.1016/J.KNOSYS.2018.01.014. URL: https://www.sciencedirect.com/science/article/pii/S0950705118300224.

[74] Sebastian Ruder, Parsa Ghaffari, and John G Breslin. "Character-level and Multi-channel Convolutional Neural Networks for Large-scale Authorship Attribution". In: *arXiv preprint arXiv:1609.06686* (2016).

[75] Chidansh Amitkumar Bhatt and Mohan S Kankanhalli. "Multimedia data mining: state of the art and challenges". In: *Multimedia Tools and Applications* 51.1 (2011), pp. 35–76.

[76] Pradeep K. Atrey et al. "Multimodal fusion for multimedia analysis: a survey". In: *Multimedia Systems* 16.6 (Apr. 2010), pp. 345–379. ISSN: 0942-4962. DOI: 10.1007/s00530-010-0182-0. URL: http://dx.doi.org/10.1007/s00530-010-0182-0%20http://link.springer.com/10.1007/s00530-010-0182-0.

[77] Deli Pei et al. "Unsupervised multimodal feature learning for semantic image segmentation". In: *The 2013 International Joint Conference on Neural Networks (IJCNN)*. IEEE, Aug. 2013, pp. 1–6. ISBN: 978-1-4673-6129-3. DOI: 10.1109/IJCNN.2013.6706748. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6706748.

[78] Oriol Vinyals et al. "Show and Tell: A Neural Image Caption Generator". In: *CoRR* (Nov. 2014). arXiv: 1411.4555. URL: http://arxiv.org/abs/1411.4555.

[79] Kelvin Xu et al. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention". In: *arXiv preprint arXiv:1502.03044* (2015).

[80] Miguel A. Álvarez-Carmona et al. "A visual approach for age and gender identification on Twitter". In: *Journal of Intelligent & Fuzzy Systems* 34.3133-3145 (2018), pp. 1–5.

[81] Michele Merler, Liangliang Cao, and John R. Smith. "You are what you tweet...pic! gender prediction based on semantic analysis of social media images". In: *2015 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, June 2015, pp. 1–6. ISBN: 978-1-4799-7082-7. DOI: 10.1109/ICME.2015.7177499. URL: http://ieeexplore.ieee.org/document/7177499/.

[82] Tomoki Taniguchi et al. "A Weighted Combination of Text and Image Classifiers for User Gender Inference". In: *Proceedings of the 2015 Workshop on Vision and Language (VL'15)*. Lisbon, Portugal, 2015, pp. 87–93. URL: http://www.anthology.aclweb.org/W/W15/W15-2814.pdf.

[83] Michele Merler, Liangliang Cao, and John R. Smith. "You are what you tweet...pic! gender prediction based on semantic analysis of social media images". In: *2015 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, June 2015, pp. 1–6. ISBN: 978-1-4799-7082-7. DOI: 10.1109/ICME.2015.7177499. URL: http://ieeexplore.ieee.org/document/7177499/.

[84] S Azam and M Gavrilova. "Gender prediction using individual perceptual image aesthetics". In: (2016). URL: https://otik.zcu.cz/handle/11025/21646.

[85] Ryosuke Shigenaka, Yukihiro Tsuboshita, and Noriji Kato. "Content-Aware Multi-task Neural Networks for User Gender Inference Based on Social Media Images". In: *2016 IEEE International Symposium on Multimedia (ISM)*. IEEE, Dec. 2016, pp. 169–172. ISBN: 978-1-5090-4571-6. DOI: 10.1109/ISM.2016.0040. URL: http://ieeexplore.ieee.org/document/7823607/.

[86] Quanzeng You et al. "The Eyes of the Beholder: Gender Prediction Using Images Posted in Online Social Networks". In: *2014 IEEE International Conference on Data Mining Workshop*. IEEE, Dec. 2014, pp. 1026–1030. ISBN: 978-1-4799-4274-9. DOI: 10.1109/ICDMW.2014.93. URL: http://ieeexplore.ieee.org/document/7022709/.

[87] Xiaojun Ma, Yukihiro Tsuboshita, and Noriji Kato. "Gender estimation for SNS user profiling using automatic image annotation". In: *2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*. IEEE, July 2014, pp. 1–6. ISBN: 978-1-4799-4717-1. DOI: 10.1109/ICMEW.2014.6890569. URL: http://ieeexplore.ieee.org/document/6890569/.

[88] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. "Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (July 2015), pp. 1556–1566. URL: http://www.aclweb.org/anthology/P15-1150.

[89] Duyu Tang, Bing Qin, and Ting Liu. "Document Modeling with Gated Recurrent Neural Network for Sentiment Classification". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (Sept. 2015), pp. 1422–1432. URL: http://aclweb.org/anthology/D15-1167.

[90] Efstathios Stamatatos. "A survey of modern authorship attribution methods". In: *Journal of the American Society for Information Science and Technology* 60.3 (2009), pp. 538–556. ISSN: 1532-2890. DOI: 10.1002/asi.21001. URL: http://dx.doi.org/10.1002/asi.21001.

[91] Diederik Kingma and Jimmy Ba. "Adam: a Method for Stochastic Optimization". In: *International Conference on Learning Representations (ICLR)* (2015).

[92] Moshe Koppel and Yaron Winter. "Determining if two documents are written by the same author". In: *Journal of the Association for Information Science and Technology* 65.1 (2014), pp. 178–187.

[93] Moshe Koppel and Jonathan Schler. "Authorship Verification As a One-class Classification Problem". In: *Proceedings of the Twenty-first International Conference on Machine Learning*. ICML '04 (2004), p. 62. DOI: 10.1145/1015330.1015448. URL: http://doi.acm.org/10.1145/1015330.1015448.

[94] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. "An Empirical Exploration of Recurrent Network Architectures". In: *Proceedings of the 32nd International Conference on Machine Learning* 37 (2015). URL: http://machinelearning.wustl.edu/mlpapers/paper%7B%5C_%7Dfiles/icml2015%7B%5C_%7Djozefowicz15.pdf.

[95] Klaus Greff et al. "LSTM: A search space odyssey". In: *IEEE transactions on neural networks and learning systems* PP.99 (2016), pp. 1–11. ISSN: 2162-237X. DOI: 10.1109/TNNLS.2016.2582924.

[96] Jiwei Li et al. "Visualizing and Understanding Neural Models in NLP". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (June 2016), pp. 681–691. URL: http://www.aclweb.org/anthology/N16-1082.

[97] Martin Potthast et al. "Improving the Reproducibility of PAN's Shared Tasks: Plagiarism Detection, Author Identification, and Author Profiling". In: *Information Access Evaluation meets Multilinguality, Multimodality, and Visualization. 5th International Conference of the CLEF Initiative (CLEF 14)*. Ed. by Evangelos Kanoulas et al. Berlin Heidelberg New York: Springer, Sept. 2014, pp. 268–299. ISBN: 978-3-319-11381-4. DOI: 10.1007/978-3-319-11382-1_22.

[98] Piotr Bojanowski et al. "Enriching Word Vectors with Subword Information". In: *arXiv preprint arXiv:1607.04606* (2016).

[99] François Chollet. *Keras*. `https://github.com/fchollet/keras`. 2015.

[100] John Arevalo et al. "Gated Multimodal Units for Information Fusion". In: *5th International conference on learning representations 2017 workshop*. 2017.

[101] Miguel A. Álvarez-Carmona et al. "A visual approach for age and gender identification on Twitter". Unpublished paper. 2017.

[102] K. Simonyan and A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR* abs/1409.1556 (2014).

[103] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252. DOI: `10.1007/s11263-015-0816-y`.

[104] Diederik Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[105] James Bergstra and Yoshua Bengio. "Random search for hyper-parameter optimization". In: *Journal of Machine Learning Research* 13.Feb (2012), pp. 281–305.

[106] Pashutan Modaresi, Matthias Liebeck, and Stefan Conrad. "Exploring the Effects of Cross-Genre Machine Learning for Author Profiling in PAN 2016". In: *CLEF*. 2016.