Style preferences in the recommendation system of articles

Ahmed Lahlou Mimi, Hamza Jeddad, Kenza Lahlali, Moustapha Ebnou, Wassim Gharbi

CentraleSupelec under the supervision of ***Julien Hay***

Abstract

Keywords: Style, Machine Learning, Recommendation

1. Introduction

Analysing literary style - also refered to as stylometry - has recently picked up interest from companies and researchers alike. Despite being a deeply rooted topic of natural language processing and machine learning, todays context where fake news are hardly identified gives the field an extreme importance. It becomes immediately obvious, for the very amateur of stylometry, that many applications can be based on established and improved methods. Among which classification of spam emails, identification of worth news, authorship attribution and verification, and news recommendation are worth noting. Style is an important element that affects the choice of the reader, and thus needs to be included in recommaendation systems [9]. In the following work, we focus on recent Machine Learning advances in Stylometry and their use for source attribution and news recommendation.

2. Definition of Stylometry

On a broader scale, a style is a way of accomplishing a task that is specific to a certain period, place, person or movement. We limit the scope of this article to literary style on which much of the interest is put from the scientific community. Among the first and most-known attempts to use style study for authorship attribution are the Federalist Papers; Alexandre Hamilton, James Madison and John hay anonymously published a set of 85 papers

to promote the ratification of the United States constitution.It was only by studying style features and word choices that the authors have been identified for most of the articles, yet 12 remained disputed. Later, more statistical studies were conducted on the Federalist and more generally on stylometry. Many have tried to define literary style. In fact, [5] defines Stylometry as the analysis of authorial style. More precisely, stylometry is divided into 5 main subtasks: Authorship attribution,Authorship Verification, Authorship Profiling, Stylochrometry and Adversal Stylometry.

- Authorship Attribution: aims to determine the probability that a document was written by a particular author. Some research papers also successfully tried to attribute a text to a group of people. (Organizations, Group affiliations) [8] .
- Authorship Verification: aims to determine the probability that two documents were written by the same author.
- Authorship Profiling: aims to determine the demographic characteristics of the author (Gender, Age etc.).
- Stylochronometry: studies the changes in the authorial style over time.
- Adversarial Stylometry: explores the ways to imitate or erase the style of an author or a document.

The applications of stylometry are numerous. For example, machine-learning based algorithms were used to detect chat bots based on style [2]
[13] defines the style as the choice of a word instead of another one, of an expression instead of another one etc.There is a wide variety of words, conjunctions, verbs, expressions and the choices made by the author characterize the style of the author.

3. Features

One of the challenges stylometry experts face is to determine what precisely defines a style and how to detect it. Usually, a set of features and metrics are selected and assumed to be a good measure of style. Those range from a variety of types.

3.1. Lexical Features

Lexical Features are used to capture stylistic traits at each level. They are generally based on words or characters. For character features, we often find n-grams. The value of n varies from one language to another, and allows to capture different levels of variables (syllables, terms etc.). For word features, the words are used directly, but there is a difficulty for some languages where the words are not delimited. Researchers have also developed variables related to vocabulary richness, but these are very dependent on language. Examples: number of words, number of characters, average sentence length etc.

3.2. Semantic Features

These features aim at grasping the meaning behind words and sentences, by using synonyms and semantic dependencies for example. We find in this category POS tagging (Part-of-speech tagging) which aims to assign to each word its semantic function, and named entity recognition, which aims to detect named entities in a text. In [13], the author based its features on :

- Function Words: these are primarily grammatical function (and, for and the), the interdependence of different function word frequencies with style will result in effective attribution.

- Systemic Functional Grammar: models languages as a system of choices (choosing ("I", "me") or ("they","them"). By analyzing all the different possibilities that the author has, we can build a system network for computational analysis of textual style

- Cohesion: refers to how a text is constructed. If it's by elaboration, the author used "in other words", "for example", etc. If it's by extension, the author uses "and", "or", "besides", etc. If it's by enhancement, the author uses "similarly", "therefore", "consequently", etc.

- Assessment: means how a text uses statements of belief, obligation, or necessity. The author expresses modality through verbs ("can", "might", "should", "must"), adverbial adjunct (probably, preferably), projective clauses (I think that..., It is necessary that...).

- Appraisal: how the text adjudges the quality of various objects or events. Indeed, the author can express appreciation ("amazing", "beautiful", "innovative"), affect ("happy", "joyful", "gloomy", "miserable") or judgment ("brave", "faithful", "generous").

3.3. Syntactic and Structural Features

Syntactic Features Captures style in the organization of sentences. It Focuses on elements like punctuation (Punctuation frequency), function words (Function word Frequency) etc. Structural Features aims to capture how an author organizes his documents. Examples: Paragraph length, indentations etc.

3.4. Entity and Topic Modeling

In [13], the author used entity-, topic- and hashtag-based features. The extraction of the topic was done thanks to *Open Calais*, which distinguishes between 18 different topics among them culture, entertainment etc. The entity-based features enabled the extraction of the entity of the tweet, for example the person, the place or the event. It was shown that the diversity of profiles gives better results. The hastag-based features wasn't successful. Entity-based features was more successful than topic-based features.

3.5. Polarization Features

In [14], the authors try to detect sarcastic tweets. In order to do that, the authors considered three types of features, each one represented by the output (a vector) of a pre-trained CNN-Network.

- Sentiment Features: This category aims to grasp the polarity of the tweet (positive, negative, neutral). Many considerations enter in the choice of this features. For example, if the tweet contains words or expressions with opposed polarity, it is a strong indication that the tweet may contain sarcasm.

- Emotion Features: The goal is the capture the emotions expressed in the tweet (Surprise, Disgust etc.). The purpose is to detect shifts in emotion to detect sarcasm.

- Personality Features: The aim is to grasp personality traits in tweets. The intuition behind this choice is that sarcasm is user specific.

In addition to a wide range of options for choosing features, developers have also many choices when it comes to the algorithm. It is worth noting that feature engineering can improve the overall performance of the full model. For instance, faced with the challenge of cross-thematic and cross-genre authorship attribution, Stamatatos [15] has introduced an algorithm of distorting the original text before extracting character or word-level n-grams. This comes as a suggested solution to the inherent problem of separating topic/genre related information from the personal style of a certain author. The goal of text distortion mechanism is to hide genre/topic related features and only keep information that is relevant to determining the writing style. One way of achieving this is by masking tokens which frequency of occurrence in a preset corpus is below a well calibrated threshold. By comparing performances on two data sets (1 topic-centered, and 1 heterogeneous), this method has indeed improve the performance on the cross-topic, cross-genre set.

Another replacement for features directly extracted from the text is to represent with GloVe. The method is an unsupervised learning algorithm for obtaining vector representations for words in a text where training is performed on aggregated global word-word co-occurrence statistics from a corpus. GloVe captures nuances by encoding meaning in co-occurrence probability matrix. It has been used for many studies as a first step before running a classifier on a text.

Thanks to the previous work on features, it was shown that when the features were simple and well defined the result was more accurate. We can then consider simple features on words in order to capture information on the personality, sentiment and emotion of the author.

4. Modelling

The information extracted for the articles can be used as input to learning algorithms.

4.1. Compression and Similarity based Models

In [10], the author presents all the different methods for **lexical similarity** and **semantic similarity**. For **lexical similarity**, there are :

- Longest common subsequence Similarity : measure the similarity between two strings

- N-gram similarity : determine the similarity of subsequence of n items from given text sequence
- Levenshtein distance similarity : computes the minimum number of operations to transform one string into another one
- Cosine similarity : the text is transformed into a vector, the cosinus similarity formula is computed.
- and other similarity measures like : Centroid based similarity, Web Jaccard Similarity, Web Simpson similarity, Web PMI similarity.

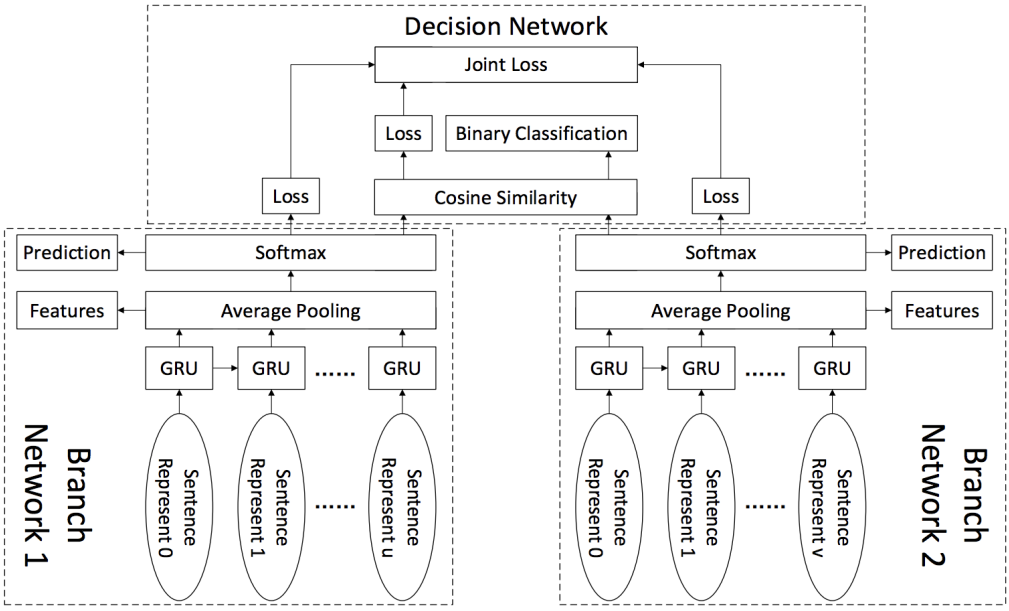
For **semantic similarity** which determines the similitude between two texts in terms of their meaning :

- Normalized Google Distance : similarity between keyword on the basis of the number of hits returned by Google Engine.
- Normalized Information Distance : computes the information distance between two texts as written in the paper.
- and other methods such as : Knowledge Based Similarity, Resnik Similarity, Vector similarity and Fusion similarity measure.

In [5], the performance of two sets of Algorithms is Analyzed. The first set consists of Authorship Verification Models: Two Minimum-distance models are tested, the first with the cosine similarity and the second with the maxmin similarity. The Similarities are not computed across all dimensions, but rather on a random portion of the dimensions multiple times before being averages. The similarities are then used to make predictions. The second set consists of Authorship Attribution models: The article cites 14 algorithms that has been implemented. We can find multiple types of models: Probabilistic models Like Nave Bayes and Augmented Nave Bayes, Distance models based on stop words and frequent words, Machine Learning Models like SVM, Compression based Models that compares the compression rate of an authors documents with and without the unlabeled text. The best performances are achieved by the Koppel11 Algorithm (87% Accuracy), an algorithm that computes distances over randomly chosen dimensions created from 4-grams. The Stamatatos07 Algorithm (92% Accuracy), that make use of Stamatatos distances. The teahan03 (98% Accuracy) Algorithm that applies the PPM text Compression Algorithm.

In [3], the authors used a siamese neural network in the context of authorship identification, as they can be used to verify whether or not two given articles are written by the same author. The idea was to rely on the article-level GRU (which will be presented later in the Neural Networks part), as its composed of two identical GRU networks working separately for two inputs. The output of the average layer is treated as the extracted features from the input (how the text will be encoded), which will be sent to a similarity measure layer to compute the similarity between the two encodings (using a distance: Euclidean distance or cosine similarity).

The authors in [3] when applied this to capture the style at the article level (alongside with the article-level GRU) they came up with 0.9 test accurracy with optimal value of (which is a parameter that connects the loss from different parameters).



Glove

4.2. Machine Learning

4.2.1. SVM

In [13], the author uses SVM for linear classification which enables sibling oppositions is a pair of relative frequencies features (number of time of appearance of one feature divided by number of time of appearance of all the

features), one of which indicates one class and the other indicates the other class. The objective is to determine for each essay which class it belongs. The author of [13] used conditionality to separate classes. For instance, if we consider Class A as expressing high RF(Median—VALUE,MODALITY TYPE/Modalization) and Class B as expressing high RF(Low—VALUE,MODALITY TYPE/Modalization). Then, when a text is in Class A, it prefers to express Median (i.e., non-extreme) values, whereas in similar situations, Class B prefers to express Low values. This may indicate that texts in Class A tend to be more cautious.

In [1], the authors used SVM classifier for authorship attribution.

4.3. Neural Networks

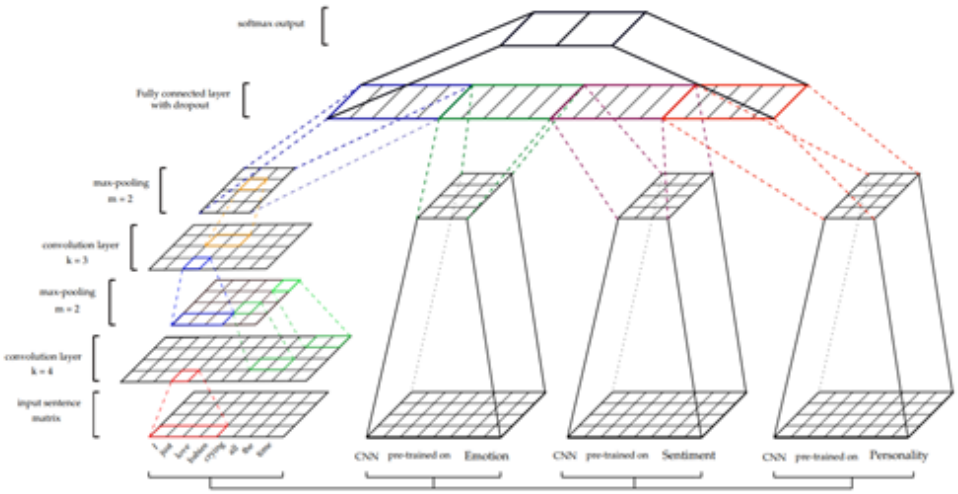
4.3.1. Stacked Denoising AutoEncoder

An AutoEncoder (AE) is an artificial neural network used in unsupervised learning. The goal of an autoencoder is to learn a representation (encoding) of a data set. It simplest form is a feedforward non-recurrent neural network having an input layer, an output layer that have the same size as the input, and one or more hidden layers, its objective is to reconstruct the input. Denoising autoencoders (DAE) take a noisy input and learn to recover the original input to avoid overfitting. Stacked Denoising AutoEncoder is a stack of many DAEs, its highest output is a representation of the original input data.[11]

In [1], the authors combined SDAE with other feature extraction methods (n-gram and bag of words) for author attribution. They obtained an accuracy over 90% for a feature size from 1000 to 10000 while the accuracy does not exceed 80% when they used only n-gram or frequency-based method.

4.3.2. Convolutional Neural Networks

In [14], the authors explored methods that use Convolutional Neural networks and applied them to Sarcasm detection. They claim to be the first article to try this approach. The authors considered that Sentiment, Emotion and personality played a crucial role in detecting sarcasm in a tweet, so they modelized each component separately using other datasets. Thus, the article proposed the following architecture, that incorporate the three types of features as independent networks (trained separately on a different dataset: Transfer Learning) and a fourth one that is directly trained with the output. The 4 networks are then connected by a Fully Connected Network (FCN).



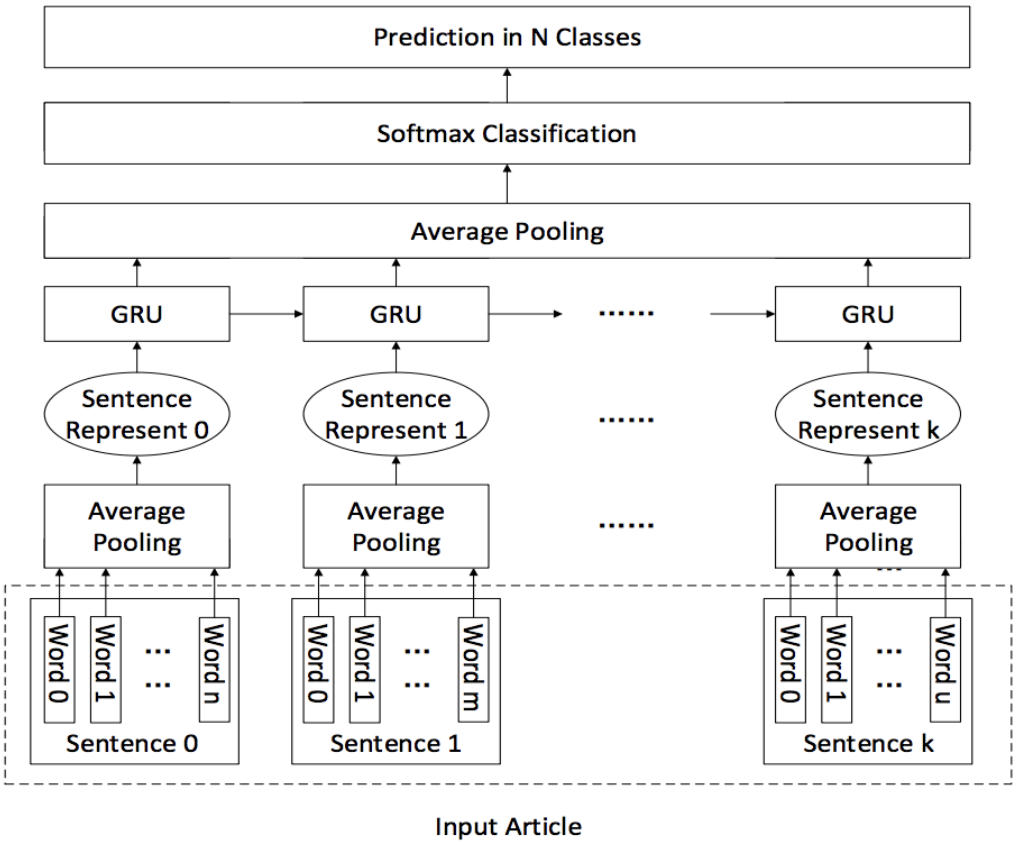
The output is directly used to predict the label through a softmax output (CNN configuration) or is used as the input of a separate SVM Model that makes the predictions (CNN+SVM configuration). The two configurations are tested on 3 Different datasets. On each one, the CNN+SVM performs better (around 94% accuracy) than the CNN (around 90% accuracy).

4.3.3. Recurrent Neural Networks

In [3], the authors used and evaluated two types of recurrent neural networks on the authorship identification performance: Gated Recurrent Unit GRU and a Long Short Term Memory LSTM. In [3] 3 different recurrent neural network models were used (alongside with a siamese neural network to examine similarity between two articles, they happen to be very powerfull in authorship identification).

- Sentence-Level GRU : The idea here is to capture the author style at the sentence level, it takes a sentence as an input, so a word is an input unit, passed through the GRU unit and does the classification on the sentence.

- Article-level GRU : This model does the same job as the first one but this time at a different scale. It takes the whole article or a bunch of paragraphs as an input and does the classification. An input unit here is a sentence, when sentences are proceeded, at each time stamp, the sentence is passed through an average pooling layer and the its average representation is passed to the GRU net.



- Article-level LSTM : It uses the same data representation and structure as for the article-level GRU, the only difference is between the two recurrent neural networks (GRU and LSTM) which is in the cell and the output gates.

- Results : The models were built and tested on two data sets (a news dataset C50 and a story dataset Gutenberg). On the sentence-Level GRU,the model was overfitting on test set, and so an article level approach was used on both GRU and LSTM models. On the article level GRU, the best accuracy for both datasets was 0.69 and 0.89 respectively and the F1 score was 0.66 and 0.85 respectively and finally on the article level LSTM: on C50 the highest accuracy was 0.62 (lower than previous model).

As mentioned above, using GloVe representation of the text can be of positive impact on the performance of the overall neural network algorithm. In

fact, RNN on pre-trained GloVe has been proved by Zhou and Wang [9] to outperform non-deep learning classifier fed with crude text or its vector representation.

5. Recommendation system

Recommending news articles is a not an easy task as the news items, which are going to be recommended, are new by its very nature.

5.1. Based on Writing style (Stylometry-based RS)

In [12] , the authors try to use stylometric features to improve an existing recommendation system for books. Their research demonstrates that writing style influences book selection and can improve results returned by existing algorihtms.

5.2. Based on User Previous activity

In [4], the author defines the profil of a user u as the weighed average of different concepts c with the weight w(u,c) and defines the vector P(u) related to the user u as P(u) = (c, w(u, c)) with c C, u U. For instance : w(u, technology) = 5 means that the user u used 5 times the hashtags with the name "technology". For each new u, the author creates P(u) and computes the similarity between P(u) and the previous P(ui) (ui the previous users) to find the closest one to P(u).

5.3. Based on external datasets (news)

In [4], in order to further enrich the semantics of Twitter messages, the authors implemented several strategies to link tweets with external Web resources. The authors evaluated these strategies and showed that they achieve 70-80% accuracy. Given the links between tweets and news articles, entities and topics extracted from articles can be propagated to the corresponding tweets to further contextualize and enhance the semantics of Twitter activities.

5.4. Collaborative Filtering Recommendation Systems

These systems are based on available ratings of active users to predict the preferences for the other users. It only requires an item-users rating matrix. It can be a User-based CF or an Item-based CF. The User-based CF searches similarity between users while Item-based CF finds similarity between two

co-rated items that are rated by the same group of users. KNN and the slope algorithm (linear regression) are the most used machine learning algorithms in this approach.[7]

5.5. Content-based RS (CB)

Content-Based Recommendation Systems (CB) use the content of the item (title, author, summary, outline, the whole text, year of publication ...) to predict the users' future preferences. The content might be represented as a bag of words, a vector or an ontology (a class of domain knowledge connected by relations).[7]

Naive Bayes, SVM, Decision Trees and KNN are used in Content-Based Systems. Many techniques are used to reduce the dimensionality of the item including Rate Adapting Poisson (RAP), Latent Dirichlet Allocation (LDA), and Probabilistic Latent Semantic Indexing (PLSA).[7] Content-based recommendation system have been used for more than news-recommendation. In fact [6] presents a book recommendation system based on authors' writing styles.

5.6. Demographic-based RS

Demographic-Based RS (DB) classify users based on age, gender, country ... and associate items with different classes. The problem with this kind of systems that can disrupt the privacy of users.[7]

5.7. Social RS

Social RS use relationships, tags and other social media content to make recommendations. Theses systems usually complement other systems to solve the new user problem.[7]

5.8. Context-aware RS

These systems exploit the current situation of the user (time, location,...) for suggestions.[7]

6. Conclusion and hints on our next work

- Features: GloVe to transform text into meaningful matrix capturing important features of the text.
- Machine learning algorithm : A combination of CNN and LSTM algorithm.

7. References

[1] Nagia Ghanem Ahmed M. Mohsen, Nagwa M. El-Makky. Author identification using deep learning. 2016.

[2] Nawaf Ali. Text stylometry for chat bot identification and intelligence estimation.

[3] Rao Zhang Chen Qian, Tianchang He. Deep learning based authorship identification.

[4] Geert-Jan Houben Ke Tao Fabian Abel, Qi Gao. Analyzing user modeling on twitter for personalized news recommendations.

[5] William Fulton. Surveying stylometry techniques and applications. 2017.

[6] Stan Szpakowicz Haifa Alharthi, Diana Inkpen. Authorship identication for literary book recommendations.

[7] Stan Szpakowicz Haifa Alharthi, Diana Inkpen. A survey of book recommender systems. 2017.

[8] Shibin Parameswaran Jeffrey Ellen. Using statistical techniques on nlp features for online group identification.

[9] Huafei Wang Liuyu Zhou. News authorship identification with deep learning.

[10] Rajesh Wadhvan Nitesh Pradhan, Manasi Gyanchandani. A review on text similarity technique used in ir and its application.

[11] Isabelle Lajoie Yoshua Bengio Pierre-Antoine Manzagol Pascal Vincent, Hugo Larochelle. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. 2010.

[12] Bruno Martins Paula Cristina Vaz, David Martins de Matos. Stylometric relevance-feedback towards a hybrid book recommendation algorithm.

[13] Moshe Koppel James W. Pennebaker Shlomo Argamon, Sushant Dhawle. Lexical predictors of personality type.

[14] Devamanyu Hazarika Prateek Vij Soujanya Poria, Erik Cambria. A deeper look into sarcastic tweets using deep convolutional neural networks. 2016.

[15] Efstathios Stamatatos. Authorship attribution using text distortion.