Graph-based and Lexical-Syntactic Approaches for the Authorship Attribution Task

Notebook for PAN at CLEF 2012

Esteban Castillo[1], Darnes Vilariño[1], David Pinto[1]
Iván Olmos[1], Jesús A. González[2], and Maya Carrillo[1]

[1]Benemérita Universidad Autónoma de Puebla
Faculty of Computer Science, Mexico

[2]Institute of Astrophysics, Optics, and Electronics
Computer Science Department, Mexico
ecjbuap@gmail.com,{darnes, dpinto, iolmos, cmaya}@cs.buap.mx
jagonzalez@inaoep.mx

Abstract. In this paper we present two different approaches for tackling the authorship attribution task. The first approach uses a set of phrase-level lexical-syntactic features, whereas the second approach considers a graph-based text representation together with a data mining technique for discovering authorship patterns which may be further used for attributing the authorship of an anonymous document. In both cases we employed a support vector machine classifier in order to determine the target class. The features extracted by means of the graph-based approach allowed it to obtain a better performance than the other approach.

Keywords: Authorship attribution, graph-based representation, phrase-level lexical-syntactic features, support vector machines

1 Introduction

Discovering the correct features in a raw text which allows unambiguously to attribute the authorship of a given anonymous document is a very hard task. In recent years, there have been a number of research papers in this direction. The traditional authorship attribution task consists of determining the correct authorship of an anonymous document, using a supervised collection of documents, i.e., a reference set of documents manually tagged with their corresponding authorship attribution. In other words, this task can be seen as a classification problem in which the target tag or class is the author name/ID.

Determining the authorship of an anonymous document is a task that has been tackled for several years by the computational linguistic community. An effort that has been empowered by the continuous growing of information in Internet. In this sense, the importance of finding the correct features for characterizing the signature or particular writing style of a given author is fundamental for solving the problem of authorship attribution.

The results reported in this paper were obtained in the framework of the 6th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse

(PAN'12). In particular, in the task named "Traditional Authorship Attribution" which has the following two sub-tasks:

- *Traditional* (closed class / open class, with a variable number of candidate authors). This subtask consists of assigning the real author name to a given anonymous document, using a set of candidate authors as reference.
- *Clustering*. A target number of paragraphs are required to be clustered into groups (between one or four) in order to obtain clusters of paragraphs that correspond to the same author.

For this purpose, we attempted two different techniques for representing the features that will be taken into account in the process of authorship attribution. The proposed approaches are discussed in the following sections.

The rest of this paper is structured as follows. In Section 2 it is presented the description of the features used in the task to be tackled. Section 3 shows the classification methods (supervised and unsupervised) employed in the experiments. The experimental setting and a discussion of the obtained results are given in Section 4. Finally, the conclusions of this research work is presented in Section 5.

2 Description of the features used in the task

In this work we explore two very different text representation schemas. The first approach considers lexical-syntactic features, whereas the second uses a data mining based process for extracting the most relevant terms of the target documents. Both schemas are described as follows.

2.1 Lexical-syntactic feature approach

In this approach are considered the following lexical-syntactic features for representing the particular writing style of a given author:

- Phrase level features
  - Word prefixes. A group of letters added before a word or base to alter its meaning and form a new word.
  - Word sufixes. A group of letters added after a word or base to alter its meaning and form a new word.
  - Stopwords. A group of words that bear no content or relevant semantics which are filtered out from the texts.
  - Trigrams of PoS. Sequences of three PoS tags[1] appearing in the document.
- Character level features
  - Vowel combination. Word consonants are removed and, thereafter, the remaining vowels are combined. Each vowel combination is considered to be a feature. Adjacent repetition of vowels are considered as only one vowel.
  - Vowel permutation. Word consonants are removed and, thereafter, the vowel permutation is considered to be a feature.

The text representation by means of the above mentioned features is described in Section 2.3.

[1] POS-tagger of NLTK

2.2 Graph-based approach

In this approach, a graph based representation is considered[5]. Given a graph $G = (V, E, L, f)$ with V being the non-empty set of vertices, $E \subseteq V \times V$ the edges, L the tag set, and $f : E \rightarrow L$, a function that assigns a tag to a pair of associated vertices. Each text paragraph is tagged with its corresponding PoS tags, in this case, using the TreeTagger tool[2]. Each word is stemmed using the Porter stemmer[3]. In this type of text representation, each vertex is considered to be a stemmed word and each edge is considered to be its corresponding PoS tag. The word sequence of the paragraphs to be represented is kept. The tag set of PoS used in the experiments is shown in Table 1.

Table 1. Description of PoS tags used

PoS tag	Description
JJ	Adjective
VBN	Verb - Past participle
WDT	Determiner
NN	Noun
CD	Number
RB	Adverb
NNS	Noun - Singular
CC	Conjuntion
RBR	Adverb - Comparative
MD	Modal
JJR	Adjective - Comparative
VBG	Verb - Present participle
VBD	Verb - Past
VBP	Verb - Present, not the 3rd person singular
VBZ	Verb - Present, 3rd person singular
FW	Unknown word
PRP	Possessive pronoun
VB	Verb in base form
NNP	Noun - Plural
RBS	Adverb - Superlative
IN	Preposition and conjunction
JJS	Adjective superlative
PDT	Predeterminer

In order to demonstrate the way we construct the graph for each phrase, consider the following text phrase: “second qualifier long road leading 1998 world cup”. The associated graph representation is shown in Figure 1.

The Subdue tool Once each paragraph is represented by means of a graph, we apply a data mining algorithm in order to find subgraphs. Subdue is a data mining tool widely used in structured domains. This tool has been used for discovering structured patterns in texts represented by means of graphs [2]. Subdue uses an evaluation model named “Minimum encoding”, a technique derived from the minimum description length principle [3], in which the best graph sub-structures are chosen. The best subgraphs are those that minimize the number of bits that represent the graph. In this case, the number of bits is calculated considering the size of the graph adjancency matrix. Thus, the best

[2] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

[3] http://tartarus.org/ martin/PorterStemmer/

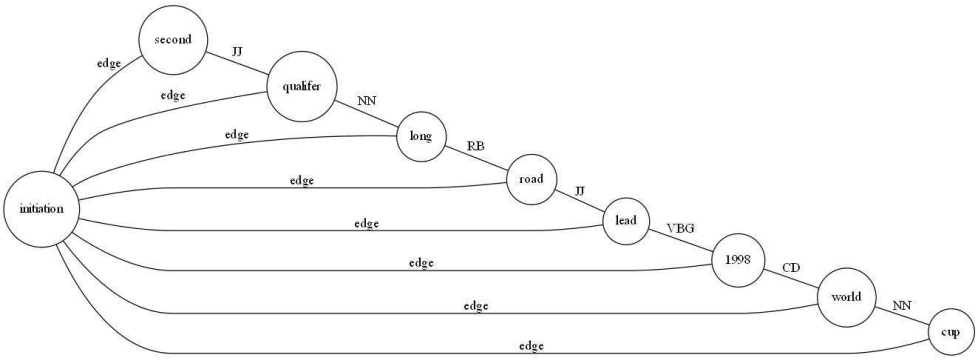

Fig. 1. Graph based text representation with words and their corresponding PoS tags

substructure is the one that minimizes $I(S) + I(G|S)$, where $I(S)$ is the number of bits required to describe the substructure S, and $I(G|S)$ is the number of bits required to describe graph G after it has been compacted by the substructure S.

2.3 Text representation schema

Let $(x_1, x_2, x_3, \cdots, x_n)$ be the set of features selected for representing the documents. Each document D is represented considering the feature frequency, i.e., we used the bag of words representation for each document[1]. It is worth noting that both approaches (lexical-syntactic and graph-based) use the same text representation schema.

The training stage uses the following feature vector:

$$D = (\underbrace{x_1, x_2, x_3, \ldots, x_n}_{Document\ features}, C) \tag{1}$$

where C is the class manually associated to the document, in this case, the author Name or ID.

For the testing stage, we use the feature vector as follows:

$$D = (\underbrace{x_1, x_2, x_3, \ldots, x_n}_{Document\ features}) \tag{2}$$

In this case, there is not a classification attribute (class name) due to the anonymous source of the document.

3 Description of the classifiers used in the task

We have used a Support Vector Machine (SVM) classifier for the problems A, B, C, D, I and J (see Section 4.1). SVM is a learning method based on the use of a hypothesis space of lineal functions in a higher dimensional space induced by a kernel, in which

the hypotheses are trained by one algorithm that uses elements of the generalization theory and taken from the optimization theory.

The linear learning machines are barely used in major real world applications due to their computational limitations. Kernel based representations are an alternative for this problem proyecting the information to a feature space of higher dimensionality which increases the computational capacity of the linear learning machines. The input space X is mapped to a new feature space as follows:

$$x = \{x_1, x_2, \ldots, x_n\} \rightarrow \phi(x) = \{\phi(x)_1, \phi(x)_2, \ldots, \phi(x)_n\} \quad (3)$$

By employing the kernel function, it is not necessary to explicitly calculate the mapping $\phi : X \rightarrow F$ in order to learn in the feature space.

In this research work, we employed as kernel the polynomial mapping, which is a very popular method for modeling non-linear functions:

$$K(x, x) = (\langle x, x \rangle + c)^d \quad (4)$$

where $c \in R$.

For problems E and F, we have employed the K-means clustering method, representing the documents with the six lexical-syntactic features previously presented. K-means is a cluster analysis method that aims to partition n observations into K clusters, considering that each observation belongs to the cluster with the closest median.

In the experiments carried out in this paper, we used the Weka data mining platform[4] for executing the implementations of SVM and the K-means classifier.

4 Experimental results

The results obtained with both approaches are discussed in this section. First, we describe the dataset used in the experiments and, thereafter, the obtained results.

4.1 Data sets

The description of the eight text collections used in the experiments (six for the traditional sub-task and two for the clustering sub-task) is shown in Table 2. As can be seen, the data set is made up of different authors. Actually, the first and second text collections (A and B) contain three different authors, the third and fourth collections (C and D) contain eight different authors, the fifth and sixth collections (I and J) contain 14 different authors, and finally, the seventh and eighth collections (E and F) contain mixed and intrusive paragraphs from 1 to 4 different authors.

4.2 Results obtained in the traditional sub-task

In Table 3 are shown the results obtained for the problems A, B, C, D, I and J of the traditional sub-task.

In order to tackle the open-class problems (B, D, and J), the training data set was enriched with documents written by unknown authors. The number of documents added

Table 2. Data set used in the experiments

Task	Problem	type	Authors	Documents training	Documents test
Traditional	A	closed-class	3	6	6
Traditional	B	open-class	3	6	10
Traditional	C	closed-class	8	16	8
Traditional	D	open-class	8	16	17
Traditional	I	closed-class	14	28	14
Traditional	J	open-class	14	28	16
clustering	E	mixed paragraph documents	1-4	2(6 paragraphs)	3 (90 paragraphs)
clustering	F	intrusive paragraph documents	1-4	2(17 paragraphs)	4(80 paragraphs)

Table 3. Results obtained in the traditional sub-task

Task	A correct/A%	B correct/B%	C correct/C%	D correct/D%	I correct/I%	J correct/J%
Graph-based approach	5/83.333	6/60	5/62.5	4/23.529	8/57.142	13/81.25
Lexical-syntactic approach	4/66.666	3/30	2/25	6/35.294	10/71.428	7/43.75

was exactly the same that the number of documents of each original collection. As may be seen in the obtained results, the best results were obtained with the graph-based representation, in which the best features were discovered with the Subdue tool. The closed-class problems obtained a better performance than the open-class ones which encourages us to investigate better methods for tackling this particular issue. It is worth noting that we retrieved almost all the authors for the open-class problem J. We consider that the number of training data is an important factor on this behavior, in other words, we consider increasing the amount of information provided by the authors.

4.3 Results obtained in the clustering sub-task

Table 4 shows the results obtained in the problems E and F of the clustering sub-task.

Table 4. Results obtained in the clustering sub-task

Task	E correct/E%	F correct/F%
Graph-based approach	68/75.555	43/53.75
Lexical-Syntactic approach	61/67.777	51/63.75

Different runs varying the number of clusters (K) were sent to the competition, however, we are not aware of the final run (the number K) reported in this paper, which was evaluated by the competition organizers. The different runs were motivated by an empirical value selected by means of the cosine similarity among different paragraphs of the text collection. This metric was a clue for determining the final number of clusters.

5 Conclusions

In this paper we presented an exploration of two different approaches. On the first hand, we employed a graph for representing text paragraphs by means of words and their corresponding part of speech tags. We aimed to consider the morphosyntactical structure

of the text (at once) for further discovering of the best features for the final representation of the training and test data. On the other hand, we evaluated a set of six lexical-syntactic features with the purpose of determining those that allow finding an appropriate signature for a given author. A higher number of features (phrase and word level) were independently evaluated, and those that provided the best discrimination scores were selected for the final evaluation.

In general, we observed that the graph-based representation obtained a better performance than the other one. However, more investigation on the graph representation is still required, so that graph patterns discovered by the Subdue tool are better than the ones obtained until now. As future work, we want to experiment with different graph-based text representations that allow us to obtain much more complex patterns.

References

1. Manning, C.D., Raghavan, P., Schtze, H.: Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA (2008)
2. Olmos, I., Gonzalez, J.A., Osorio, M.: Subgraph isomorphism detection using a code based representation. In: FLAIRS Conference. pp. 474–479 (2005)
3. Rissanen, J.: Stochastic Complexity in Statistical Inquiry Theory. World Scientific Publishing Co., Inc., River Edge, NJ, USA (1989)
4. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Series in Data Management Sys, Morgan Kaufmann, second edn. (June 2005)
5. Yan, X., Han, J.: gspan: Graph-based substructure pattern mining. In: ICDM. pp. 721–724 (2002)