A Graph Based Authorship Identification Approach
Notebook for PAN at CLEF 2015

Helena Gómez-Adorno[1], Grigori Sidorov[1], David Pinto[2], and Ilia Markov[1]

[1]Center for Computing Research,
Instituto Politécnico Nacional, Mexico
helena.adorno@gmail.com, sidorov@cic.ipn.mx, markovilya@yahoo.com
[2]Faculty of Computer Science,
Benemérica Universidad Autónoma de Puebla, Mexico
dpinto@cs.buap.mx

Abstract The paper describes our approach for the Authorship Identification task at the PAN CLEF 2015. We extract textual patterns based on features obtained from shortest path walks over Integrated Syntactic Graphs (ISG). Then we calculate a similarity between the unknown document and the known document with these patterns. The approach uses a predefined threshold in order to decide if the unknown document is written by the known author or not.

1 Introduction

Authorship verification is a problem related to authorship attribution, and can be described as follows. Given a set of documents written by a single author and a document in question, the goal is to determine if this document was written by this particular author or not [2]. It is a variant of the general authorship attribution problem with binary classification of authors: yes or no.

This task is more complex than the Authorship Attribution, because the training set is smaller, and it can be composed by only one document. Therefore, it cannot be solved as a supervised classification problem, where we usually need a greater training set. There are two categories for an author verification method, *intrinsic* and *extrinsic*. *Intrinsic* methods only use the known texts and the unknown text of each problem to decide whether they are written by the same author or not. *Intrinsic* methods do not need any other texts by other authors. *Extrinsic* methods required additional documents from external sources written by other authors. The methods use these documents as negative samples for each problem. The majority of the methods presented at PAN'14 falls into the *intrinsic* category [7], however, the winning system of PAN'13 belongs to the *extrinsic* category [8].

Our approach falls in the intrinsic category, and it uses a model for representing texts by means of a graph, the Integrated Syntactic Graph (ISG) [3], and then extracts features for similarity calculation. The ISG is built using linguistic features of various levels of language description, which provide important information about the writing style of authors. The similarity is performed between an "unknown-author" document and the "known-author" document for each problem of the evaluation corpus. If the

"unknown-author" document exceeds a predefined threshold, then it is written by the author of that problem.

The rest of this paper is structured as follows. Section 2 presents a brief description of the Integrated Syntactic Graph representation, the process for the feature extraction and the similarity calculation algorithm. Section 3 shows the proposed approach (unsupervised algorithm) used in the experiments. The experimental setting and a discussion of the obtained results are given in Section 4. Finally, conclusions are presented in Section 5.

2 Integrated Syntactic Graph

The Integrated Syntactic Graph (ISG) is a textual representation model proposed in [3] with the aim to integrate into a single data structure multiple linguistic levels of natural language description for a given document. This model is able to capture most of the features available in a text document, from the morphological to the semantic and discoursive levels. By including lexical, syntactic, morphological, and semantic relations into the representation, the model is capable to integrate in the ISG various text components: words, phrases, clauses, sentences, etc.

A complete description of this representation model is given in [3]; however, for better understanding of the application of such model in the authorship identification task, we summarize the construction process.

The construction of the ISG starts by analyzing the first sentence of the target text. We apply the dependency parser in order to obtain the parsed tree of the first sentence. This tree has a generic node (named ROOT), to which the rest of the sentences will be attached in order to form the representation of the complete graph. We perform similar actions for the second sentence in the text, applying the dependency parser and attaching the obtained syntactic tree to the ROOT node. The repeated nodes of new trees are collapsed with the identical existing nodes. In this way, we create new connections between nodes (containing the same lemmas and POS tags) of different sentences that would not exist otherwise.

The collapsed graph of three sentences is shown in Figure 1; each node of the graphs is augmented with other annotations, such as the combination of lemma (or word) and POS tags (lemma POS). Each edge contains the dependency tag together with a number that indicates the frequency of that dependency tag plus the frequency of the pair of nodes, both calculated using the occurrences in the dependency trees associated to each sentence.

2.1 Feature Extraction from ISGs

This representation allows to find features in the graph in two principal ways: (1) counting text elements (lemmas, PoS tags, dependency tags), and (2) constructing syntactic n-grams [5] while shortest paths are traversed in the graph. For this research work we used the first one.

Let us consider the first three sentences of a given text: "*I'm going to share with you the story as to how I have become an HIV/AIDS campaigner. And this is the name*

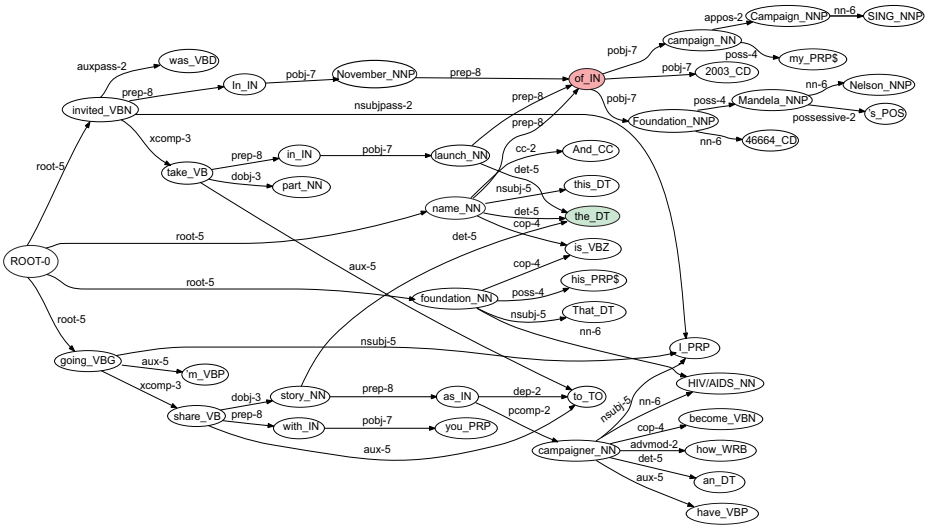

Figure 1. Example of the Integrated Syntactic Graph for three sentences

of my campaign, SING Campaign. In November of 2003 I was invited to take part in the launch of Nelson Mandela's 46664 Foundation". From these sentences we can built the ISG shown in Figure 1, where there are different paths connecting the node $ROOT - 0$ with the node of_IN. We take the shortest path that has the following features at different levels of the language description:

- Lexical level: $ROOT$, $name$, of.
- Morphological level: NN, IN.
- Syntactic level: $root$, $prep$.

A text represented by a ISG provides a set of features for each of the shortest paths found in this graph. So, for example, for construction of a vector space model representation of the document, we can consider each path as a vector of linguistic elements with numeric values (frequencies).

The feature extraction procedure starts by selecting the root node of the graph as the initial node for the path traversal, whereas the final nodes correspond to the remaining nodes of the graph reachable from the initial node. We use the Dijkstra algorithm [1] for finding the shortest path between the initial and the final nodes. While traversing the paths, we count the occurrences of all multi-level linguistic features considered in the text representation. For example, given the pair ($ROOT - 0$, to_TO), there are three ways to reach the node to_TO from $ROOT - 0$, but the shortest one is: $ROOT - 0$, $invited\_VBN$, $take\_VB$, to_TO. We count the linguistic information from that path, storing it in a pair-feature matrix.
So, considering the ISG shown in Figure 1, the vector of features
$v = invited, talk, \cdots, nam, VBN, VB, \cdots, NN, xcomp, aux, \cdots, det$
will have the following values

$v = 0, 0, \cdots, 1, 0, 0, \cdots, 0, 0, 0, \cdots, 0$ for the path $ROOT - 0$ to of_IN and $v = 1, 1, \cdots, 0, 1, 1, \cdots, 0, 1, 1, \cdots, 0$ for the path $ROOT - 0$ to to_TO

2.2 Similarity Calculation

In order to compute text similarity, we first build the ISGs for the two compared documents, and then obtain the textual patterns for each document, which gives a set of m feature vectors $\overrightarrow{f_{t,i}}$ for each text t.

The idea is to search for occurrences of features of a test document (i.e., a document of the unknown authorship (for the authorship identification task)) in a much larger graph (a graph of documents of the known authorship (for the same task)). In a graph corresponding to one author, we collapse all documents written by the author, and, therefore, it contains all the characteristics of this specific author.

Thus, the unknown author's graph D_1 is represented by m feature vectors $D_1^* = \{\overrightarrow{f_{D1,1}}, \overrightarrow{f_{D1,2}}, \cdots, \overrightarrow{f_{D1,m}}\}$, and the known author's graph D_2 by feature vectors $D_2^* = \{\overrightarrow{f_{D2,1}}, \overrightarrow{f_{D2,2}}, \cdots, \overrightarrow{f_{D2,m}}\}$. Here, m is the number of different paths that can be traversed in both graphs, using the ROOT-0 node as the initial node, while each word appears in the unknown author's graph as the final node.

Once we obtain the vector representation of each path for a pair of graphs, we adapt the cosine measure for determining the similarity between the unknown document D_1 and the known document D_2, using the cosine similarities between paths:

$$
\begin{aligned}
Similarity(D_1^*, D_2^*) &= \sum_{i=1}^{m} Cosine(\overrightarrow{f_{D_1,i}}, \overrightarrow{f_{D_2,i}}) \\
&= \sum_{i=1}^{m} \frac{\overrightarrow{f_{D_1,i}} \cdot \overrightarrow{f_{D_2,i}}}{||\overrightarrow{f_{D_1,i}}|| \cdot ||\overrightarrow{f_{D_2,i}}||} \\
&= \sum_{i=1}^{m} \frac{\sum_{j=1}^{|V|} (f_{(D_1,i),j} \times f_{(D_2,i),j})}{\sqrt{\sum_{j=1}^{|V|} (f_{(D_1,i),j})^2} \times \sqrt{\sum_{j=1}^{|V|} (f_{(D_2,i),j})^2}},
\end{aligned}
$$

where V is the total number of linguistic features.

3 Authorship Verification Approach

We follow the same approach for the English, Spanish and Dutch languages, but for the Greek language we made a modification in the methodology due to the lack of a free syntactic parser for this language.

For each problem we concatenate the "known-author" documents and represent them with an Integrated Syntactic Graph (ISG) [3] as described in the previous section. After this, the "unknown-author" documents of each problem are individually represented with an ISG using the same features. In this way, we obtained one ISG for each "unknown-author" document. In order to identify if the "unknown" document corresponds to the author of the problem in question, we calculate the similarity of that

"unknown" document (graph) with the "known-author" graph of the problem. If the similarity is greater than a predefined threshold, then the answer is "yes", i.e., it belongs to this author. However, if the similarity is lower than the predefined threshold, then the answer is "no" (it does not belong to this author). The threshold is currently obtained from the training set by averaging the similarities scores of all problems. The threshold is fixed for the complete evaluation corpus.

We decided to give an answer for all the problems, and to use the probability scores "0" when the document does not correspond to the author of its problem and "1" if the document belongs to the author of its problem.

In order to implement our approach, we used several linguistic tools in order to perform the syntactic and morphological analysis. We used the Stanford parser [1] for the English corpus, the Freeling tool [2] for the Spanish corpus, the Alpino Parser[3] for the Dutch corpus and AUEB's POS tagger[4] for the Greek corpus.

The implementation of the authorship verification system for the Greek corpus differs from the others only in the ISG representation, because it does not use the syntactic information. Instead we used a fixed graph topology, where each sentence of a document is represented by a lineal tree. We defined a $ROOT$ node for each document and all the sentences in the document are attached to the $ROOT$ node. The nodes are composed by the word concatenated with its POS tag, and if this combination is repeated in a document, the nodes are collapsed in the same way as explained in section 2. The rest of the approach remains the same.

4 Experimental Results

Table 1 presents the results obtained by our approach for each of the data sets. The best performance was obtained for the Dutch data set followed by the Greek data set. The English and Spanish data sets obtained equal performance. The results of the other participants and the description of the evaluation corpus can be found in [6].

It can observed that our results are low in comparison to the rest of the participants, but it is necessary to take into account that our approach only compares one unknown-author document against one or more known-author documents. Our approach does not require any external information, and the runtime depends mainly on the performance of the used linguistic tools. It can be observed that the runtime of the Greek corpus is significantly lower in comparison with the other three languages. This runtime difference is mainly due to the linguistic tools, given that for the Greek language we do not perform syntactic analysis.

5 Conclusions

In this paper, we described a graph based approach for the Authorship Verification task. Our approach only uses information about the texts in the given corpus and does

[1] http://nlp.stanford.edu/software/lex-parser.shtml
[2] http://nlp.lsi.upc.edu/freeling/
[3] http://www.let.rug.nl/vannoord/alp/Alpino/
[4] http://nlp.cs.aueb.gr/software.html

Table 1. Results obtained in the different languages

Language	AUC	c@1	Final Score	Runtime
English	0.53	0.53	0.2809	07:36:58
Spanish	0.53	0.53	0.2809	00:50:40
Dutch	0.62452	0.62452	0.38985	83:58:15
Greek	0.59	0.59	0.3481	00:09:21

not need any external information. The runtime is greater than the rest of the systems because of the use of several linguistic tools, such as syntactic parser and morphological tagger. The evaluation results are among the average between the rest of the participants, but we believe that this can be improved.

In order to improve our results, we need to implement an algorithm for obtaining a confidence score for the answers, instead of answer only "1" and "0" as we did in this version of the system. We also need to perform more experiments in order to determine the exact configuration of the graph representation to be used for a given corpus. Additionally, we are planning to evaluate the performance of the soft cosine measure [4] for this task. Finally, in order to decrease the runtime we can implement parallel computing.

Acknowledgments. This work was done under partial support of the Mexican Government (CONACYT PROJECT 240844, SNI, COFAA-IPN, SIP-IPN 20151406, 20144274) and FP7-PEOPLE-2010-IRSES: WIQ-EI, European Commission project 269180.

References

1. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik 1, 269–271 (1959)
2. Koppel, M., Schler, J., Bonchek-Dokow, E.: Measuring differentiability: Unmasking pseudonymous authors. J. Mach. Learn. Res. 8, 1261–1276 (Dec 2007)
3. Pinto, D., Gómez-Adorno, H., Vilariño, D., Singh, V.K.: A graph-based multi-level linguistic representation for document understanding. Pattern Recognition Letters 41(0), 93 – 102 (2014)
4. Sidorov, G., Gelbukh, A.F., Gómez-Adorno, H., Pinto, D.: Soft similarity and soft cosine measure: Similarity of features in vector space model. Computación y Sistemas 18, 491–504 (2014)
5. Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., Chanona-Hernández, L.: Syntactic n-grams as machine learning features for natural language processing. Expert Systems with Applications 41(3), 853 – 860 (2013)
6. Stamatatos, E., Daelemans, W., Verhoeven, B., Juola, P., Lopez, A.L., Potthast, M., Stein, B.: Overview of the author identification task at pan 2015. In: Working Notes Papers of the CLEF 2015 Evaluation Labs, CEUR Workshop Proceedings (2015)
7. Stamatatos, E., Daelemans, W., Verhoeven, B., Potthast, M., Stein, B., Juola, P., Sanchez-Perez, M.A., Barrón-Cedeño, A.: Overview of the author identification task at pan 2014. In: Working Notes for CLEF 2014 Conference. vol. 1180, p. 31 (2014)
8. Stamatatos, E., Joula, P.: Overview of the author identification task at pan 2013. In: Working Notes Papers of the CLEF 2013 Evaluation Labs (2013)