Tree Classifiers for Authorship verification

Yassine Landa

March 29, 2016

Introduction:

Because the increase of text in digital form and the need to intelligent systems that manage it, fields of study such as Author Authentication has gained more interests in the last years. The applications of such techniques are different and very useful in different traditional areas (e.g finding the truth in the case of disputed novels, identification of authors who publish documents anonymously) and applied domains (e.g for law and journalism the identification of a document such as a ransom note may be crucial to save lives, identification of bullying message on social networks, identification of terrorists allegations).

This techniques are based on information retrieval and computational linguistic. They assume that each author have a linguistic signature that can be traced between the lines. Different interesting elements can be found from the style of a written document such as general characteristics and personality traits.

The rise of internet and the anonymity it provides stimulated also the developpment of this field. It is often important from a financial or legal point of view to determine is two documents were in fact written by a single author or not:

- The aim might be to prevent and detect email impersonation through compromised accounts.
- We might need to do alias classification which became a major issue in cyber forensics. In fact, anonymization techniques like Tor's onion routing have made it very difficult to trace identities of suspects.
- Authorship attribution can be used to enhancing services such as search engines and recommending systems.
- We might also want to know if several tendentious product reviews were actually by the same, possibly self-interested, writer.

the PAN (http://pan.webis.de) competitions at CLEF (Conference and Labs of the Evaluation Forum) are organized each year, and are what is driving this field of studies for the last decade. In this paper we will use the English corpus proposed for the 2015 edition to solve the Author Verification task using Decision Trees and Random Forest methods. At first we will talk about the different methods and algorithmic approaches used to solve the Author Identification problem. After that we will discuss the document represented chosen for the English corpus of PAN 2015. Finally we will explain the experiment and analyse the results.

The size of the corpus of PAN 2015 matches the size of previous editions. And cross-genre and cross-topic verification cases were considered. This constitute a much more challenging and realistic case. The main goal is to find whether authorship verification methods are heavily affected by variations in genre and topic among the documents of a verification case.

Related Studies

One form of the author attribution problem that have been studied a lot is the attribution of an anonymous text to one of the authors from a small candidate set. Furthermore, it usually assumed that we have a descent amount of text by each candidate author and that the anonymous text is relatively long. But in the real world, that is not usually the case. A common situation is when we have a large list of candidates and in which we have no guarantee that the real author is part of the candidates. In those cases, it is interesting to reformulate the problem of author attribution to a series of author verification tasks. The author identification problem can be described in various ways, depending on the set of candidate authors (closed or open) and their number. One particular variation is the task of author verification that can be seen as a one-class classification problem, where the set of known texts by an author constitute the target class. For example, in forensics experts are interested in identifying the author given a small number of suspects and, at the same time, be sure that the author is not someone else not under investigation. For author verification tasks we try to answer the question: “Did candidate X write the document Y?”.

In the literature automated techniques for authorship attribution can be divided into two main types: Machine-learning methods, where the known documents of an author are used as a training set to construct a classifier [3][5]. After this first step called learning, that classifier can be used to classify anonymous documents. Some of the algorithms used are multivariate analysis, decision trees/random forests and neural nets even if a good number of studies shown that linear separators works well for text categorization like Naive Bayes for the two-class problem, Winnow and Exponential Gradient and linear support vector machines [3]. The second type of methods are called Similarity-based methods which rely on some metric to measure the distance between two documents [4] [2]. The anonymous document is then attributed to that author to whose known writing (considered collectively as a single text) it is most similar.

Another way of classifying The author identification techniques is to consider these intrinsic and extrinsic models. First the intrinsic verification models which are only based on the set of documents of known authorship and the document of unknown authorship to make their decision. Second, the extrinsic verification models that make use of external resources. Documents of other authors from the training set or crawled from the internet are added to the initial corpus. The attempt here is to transform the one-class classification problem to binary or multi-class classification problem [4].

Features and Classifier Selection

The first thing that confront any author identification task the classical text mining question: How to represent the documents? Before the texts can be compared or classified they must be represented in a suitable space. This space is often built using stylometric and semantic features that can be used to transform the documents into vectors, using various models.

For the sake of this task (cross-genre and cross-topic corpus) different categories of word-set and stylistic features were used:

- **8-grams**: a token-level feature (e.g a space-free character 8-gram is (a) a string of characters of length four that includes no spaces or (b) a string of eight or fewer characters surrounded by spaces) that are vastly used for the authorship attribution task [2] [6] [3] [4] [1] because they have the advantage of being measurable in any language without specialized background knowledge [2] [4]. A t-idf model was considered for this feature.
- **Punctuation frequency** : where each document is represented by the average of punctuation marks like colon (:) comma (,) semicolon (;) exclamation mark (!) and the parentheses () per sentence.
- **Vocabulary Strength**: We tried to capture the vocabulary strength of an author by calculating the ratio: words number of stop words in text / number of total words in text.
- **Words per sentence**: a document is described by the average and standard deviation of the number of words per sentence.
- **POS Frequency**: In this feature, we try to capture the tendency of an author to use a sequence of a particular types of POS that appear more frequently than the others, if there is any. So, we calculate the frequencies of each 4-POS tag from texts and compare the known and unknown texts based on that.
- **Starting POS Frequency**: We try to list the POS tags that the author uses in the beginning of sentences according to their frequency and then compare them among the known and unknown documents to find a lexical pattern. For example, a particular author might have the tendency to start sentences with auxiliary verbs (example) or prepositions (in, for) unknowingly for a considerable number of sentences in the corpus. The feature also indicates the writing style of the author.
- **Finishing POS Frequency**: This feature is the same as the previous one except that it takes into account the tags of the words used at the end of a sentence.

As it is impossible to decide manually which of these features are most important or relevant to our problem structure, we decided to go for Decision Tree based classifier. Such classifiers are fast to train and easy to evaluate and interrupt and moreover non-parametric and for the very reason, we don't have to worry about the outliers or whether the data is linearly separable or not. The main disadvantage is that they easily over-fit, but that's where ensemble methods like Random Forest come in. The main advantages of such approaches are:

- Almost always have lower classification error and better F-scores than decision trees.
- Almost always perform as well as or better than SVMs, but are easier to understand.
- Deal really well with uneven data sets that have missing variables.
- Give a good idea of which features in the data set are most important.
- Generally train faster than SVMs.

The Experience

Figure 1 below illustrates the basic step-by-step architecture of our training script, respectively. We were given data-sets in English which contain numbers of known and unknown text samples by several authors. In a single author subset((problem), we had one or more documents that are known to be written by that author and an unknown one for which the authorship is not known. The task is to predict whether that particular unknown document is written by the same author of that author subset or not. A .json file contains the language and the problem titles of a particular language data-set. For training sets, we were given a .txt file with the answers to those subset problems, i.e. whether or not those unknown files of a subset problem are written by the same author of other files in that subset.

At first, we read the contents from the text files and used The TreeTagger library to tag the contents of files to obtain the root words and POS tag for each word. storing in the process the two first and two last tags of a sentence in separate files. Next, we counted the tf-idf (Term Frequency-Inverse Document Frequency) of each 8-gram, all the frequencies of 4-POS tag, the starting POS tags and the last POS tags from the tagged output. Also, the vocabulary strength of an author, i.e. the ratio of number of different types of root words to number of total words occurred is another feature that we considered in our approach. The average and deviation of the number of words per sentence and punctuation counts are calculated from the plain text, i.e. raw file contents. After that a metric is choosen for every feature to calculate the difference between the know and unknown documents for each problem. The training matrice (Numpy matrice) obtained at the end is saved on a file. We used the scikit.learn python library to train a Decision Tree and a Random Forest classifier with 100 trees using the features that we extracted from the training data-set on the file written .

For the case of the Random Forest Classifier we operate a grid search using Scikit Learn library to find the optimal parameters. The best model is chosen via a 10 fold cross validation

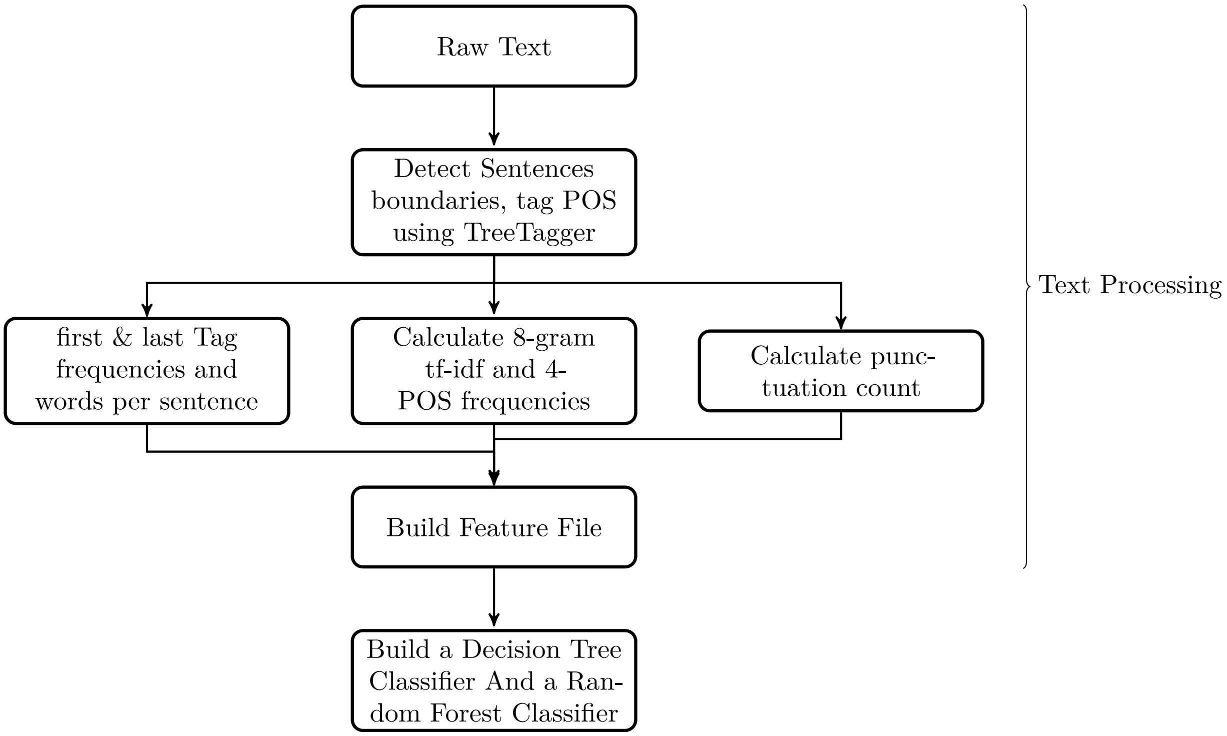
Raw Text
Detect Sentences boundaries, tag POS using TreeTagger
first & last Tag frequencies and words per sentence
Calculate 8-gram tf-idf and 4-POS frequencies
Calculate punctuation count
Build Feature File
Build a Decision Tree Classifier And a Random Forest Classifier
Text Processing

Figure 1: Basic step-by-step architecture of the training scripts

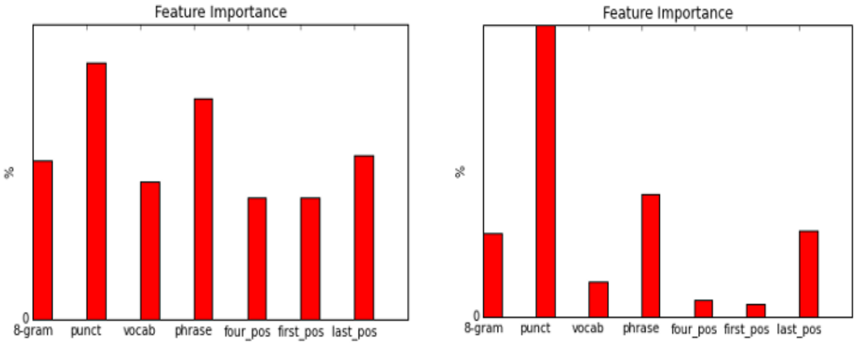

Figure 2: Feature Importance: Decision Tree Classifier (left), Random Forest Classifier (right)

process.

Results and analysis:

Table 1 shows the detailed results of the two systems in PAN 15 authorship verification task as provided by PAN organizing committee. Our present approach achieved 49fluctuation in result might be because of the variable number and size of the ‘known’- documents. To deal with multi-genre, we trained our system to analyze multiple corpuses/genre-specific training data and build the trainer as a whole.

We also looked at the most important features for each classifier. Figure 1 shows the difference of the importance of the features for the two classifiers. 8-grams, punctuation frequency, words per sentence and the finishing POS frequency features are the most important. We observe that the use of random forests stresses the difference between the important and not important features.

Conclusion and Future Scope

In this paper, we have presented an approach to solve the automatic verification task using methods of Text Analysis. It uses word based and stylistic features and two classifiers based on Decisions Trees. One Standard Decision tree classifier and a random forest classifier to classify the unknown documents based on the features extracted. In the recent years, the practical applications for authorship attribution have grown in areas such as intelligence (linking intercepted messages to each other and to known terrorists), criminal law, civil law and computer security (tracking authors of computer virus source code). This activity is part of computer science for identifying technologies, including bio-metrics, cryptographic signatures, intrusion detection systems and others.

In our future work, the accuracy of system can be improved by including some language specific features. On the other hand, the features like the average paragraph length, average word length along with a Deep Learning classifier might produce interesting hike in our system score.

References

[1] J. Frery, C. Largeron, and M. Juganaru-Mathieu. Author identification by automatic learning. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 181–185, Aug 2015.

[2] Moshe Koppel, Jonathan Schler, Shlomo Argamon, and Yaron Winter. The “fundamental problem” of authorship attribution. *English Studies*, 93(3):284–291, 2012.

[3] Moshe Koppel, Jonathan Schler, and Elisheva Bonchek-Dokow. Measuring differentiability: Unmasking pseudonymous authors. *J. Mach. Learn. Res.*, 8:1261–1276, December 2007.

[4] Moshe Koppel and Yaron Winter. Determining if two documents are written by the same author. *Journal of the Association for In-*

formation Science and Technology, 65(1):178–187, 2014.

[5] Kim Luyckx and Walter Daelemans. Authorship attribution and verification with many authors and limited data. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 513–520, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[6] Hans van Halteren. Linguistic profiling for author recognition and verification. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.