Authorship Verification by combining SVMs with kernels optimized for different feature categories

Notebook for PAN at CLEF 2015

Julián Solórzano[1], Víctor Mijangos[1], Alejandro Pimentel[1], Fernanda López-Escobedo[2], Azucena Montes[1], and Gerardo Sierra[1]

[1] Grupo de Ingeniería Lingüística, Instituto de Ingeniería, UNAM, Mexico City, Mexico
{jsolorzanos, vmijangosc, amontesr, gsierram}@iingen.unam.mx, pi.p15@hotmail.com
[2] Licenciatura de Ciencia Forense, Facultad de Medicina, UNAM, Mexico City, Mexico
flopeze@unam.mx

Abstract We present our approach to the PAN-2015 authorship verification task. We combine one-class SVM classifiers under the hypothesis that different categories of features a) are better suited for different authors and b) have different underlying topologies. Thus, we have each classifier operate in a different feature subset with a different kernel function, and the output is used to train a logistic regression model which assigns a different weight to each category of features. Results show that further improvement of the method is needed, and we discuss its shortcomings.

Keywords: authorship verification, one-class SVM, kernel selection, logistic regression

1 Introduction

This paper presents our approach to the Author Verification task in PAN at CLEF 2015. Author Verification is one several of authorship analysis tasks, in which it must be determined whether a given text was written or not by a certain author [8].

In the present task, a single problem consists of a set of documents, one of which is labeled as $unknown$ and the rest are labeled as $known$. There can be a total of up to 6 documents in a single problem. The task consists in determining whether the document labeled as $unknown$ is written by the same author as the rest of the documents. There are four different sets of problems, one for each of the following languages: Spanish, English, Greek and Dutch.

2 Methodology

The main idea behind the methodology we present is to train classifiers that are to become experts at analyzing features from a specific category of features, and then combine the knowledge of all these classifiers. According to [1], this is one of the two approaches to multi feature-set techniques that have been used in stylistic analysis. We

hypothesize that an ensemble classifier will automatically assign more weight to each author's distinctive feature subsets without having to try various feature combinations.

Finally, there is also the observation that not all feature spaces are necessarily equal and that different feature subspaces may have different underlying topologies. Learning the best distance metric leads to improvements in distance-based classification schemes [9]. In this work we experiment with different kernel functions instead of distance functions.

2.1 Document Representation

Tagging Documents are tokenized and the Part of Speech (POS) tag of each token is obtained. Table 1 indicates the softwares we used to do this processing according to each language.

Table 1. Taggers

Language	Software
Spanish	Freeling [5]
English	Freeling
Dutch	TreeTagger [6][7]
Greek	Greek POS Tagger[1]

Style Features For each document we obtain features in the following categories:

- **Punctuation marks** - tokens recognized as punctuation marks by the corresponding tagger
- **Multi word terms** - tokens made up of more than one word (as determined by the tagger, in this case only obtained for Spanish and English)
- **Lexical features** - Class intervals of word and sentence lengths
- **Start of sentence word profile** - the grammatical category of the word at the beginning of sentences
- **End of sentence word profile** - the grammatical category of the word at the end of sentences
- **Function words n-grams** - 1-grams, 2-grams and 3-grams
- **Function words skip-grams** - 2-grams and 3-grams, with up to 2 gaps
- **POS full tags n-grams** - 1-grams, 2-grams and 3-grams
- **POS abbreviated tags n-grams** - 1-grams, 2-grams and 3-grams. This only applies for languages in which the tagger outputs detailed POS tags in the first place (Spanish and English in this case). The abbreviated POS tag contains the grammatical category without further details such as number, gender, tense etc.
- **Character ngrams** 2-grams and 3-grams

[1] `http://php-nlp-tools.com/blog/category/greek-pos-tagger/`

All these features are used to create the vector representation of the document, i.e. a vector whose elements are the relative frequencies of each feature in that document (the frequency is relative to the total number of features in the category). A maximum of 200 features of each category was taken into account for a single document.

Distance-to-the-average An additional representation for each document is created as follows. First we compute a vector v_{avg} which contains the average frequency of all the features among all documents in the dataset. This vector is broken down into n subvectors such that each one contains the features of one of the n feature categories. Then for each document we similarly break down its frequency vector into n subvectors and obtain their distance to the corresponding subvectors of v_{avg}.

The resulting matrix encodes information about how a document deviates from the mean in each feature category, similar to Burrow's Delta [2] (only he used z-scores to normalize).

2.2 Classification

The classification is done by an ensemble that has its votes combined by means of a logistic regression model. The ensemble is comprised by n one-class SVM classifiers, each one using the features from one of the n feature categories, as well as an additional classifier that works with the distance-to-average representation of the documents. So, there is a total of $n + 1$ classifiers.

We separate each problem matrix into n feature category matrices (plus the distance-to-average matrix). In these matrices, each document represents a point on a space X. So, $d_1, ..., d_n \in X$, where $d_i, i \in \{1, ..., n\}$ are row vectors of each feature matrix representing a document. We assume points in this space are close to each other when written by the same author. Given a new point in this space, we want to determine if it is part of the cluster or it lies outside.

Dimensionality Reduction For the case of the feature frequency matrices, their high-dimensionality makes them difficult to process. We perform dimensionality reduction by taking their first two eigenvectors, which correspond to the two highest eigenvalues; i. e. the eigenvectors with highest variance. To create the new matrices we use these eigenvectors as columns; so our new data set then is in $\mathbb{R}^2$.

This not only reduced data dimensionality, but also filtered noise. We empirically noted that the performance of the experiments increased by considering only these eigenvectors.

Novelty Detection through One-Class SVM To apply novelty detection using a one class SVM, we first think of a map $\phi : X \to \mathcal{H}$, where $\mathcal{H}$ is a dot product space such that we can evaluate the dot product in the image of ϕ by a kernel function:

$$k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) \tag{1}$$

We need to detect a neighborhood of the data points such that given a new document of the same author it lies inside this neighborhood. To do this, the one-class SVM finds a function f that returns +1 if the point lies into the neighborhood and returns -1 otherwise.

The value of the function evaluated at a new point y is obtained considering in which part of the hyperplane it falls on. So, we need to separate the data set from the origin, solving:

$$\min_{w \in \mathcal{H}, \zeta \in \mathbb{R}^n, c \in \mathbb{R}} \frac{1}{2}||w||^2 + \frac{1}{v^n} \sum_i \zeta_i - c \tag{2}$$

Where $v \in (0, 1)$ and $w \cdot \phi(x_i) \leq c - \zeta_i, \zeta_i \leq 0$. This way, we can turn it into a decision function:

$$f(x) = sgn((w \cdot \phi(x)) - c) \tag{3}$$

such that it will be positive for the points in the data set; here the term $||w||$ is a support vector type regularization. Deriving the dual problem with the kernel function showed in (1), the solution for a the new point y has a support vector expansion:

$$f(y) = sgn(\sum_i \alpha_i k(x_i, y) - c) \tag{4}$$

Where x_i with $\alpha_i \neq 0$ are support vectors. We calculate the one-class SVM for each feature matrix, obtaining a total of n outputs or judges. For each feature matrix we use different kernel functions for equation 4. One of the simplest is the linear kernel:

$$k(x_i, x_j) = x_i^T x_j + c \tag{5}$$

The gaussian kernel is defined as:

$$k(x_i, x_j) = \exp(-\frac{||x_i - x_j||^2}{2\sigma^2}) \tag{6}$$

A sigmoid kernel is defined as:

$$k(x_i, x_j) = \tanh(\alpha x_i^T x_j + c) \tag{7}$$

Finally, the polynomial kernel:

$$k(x_i, x_j) = (\alpha x_i^T x_j + c)^m \tag{8}$$

These different kernels functions correspond to each data distribution and were selected in the training process. We select one kernel for each feature matrix by evaluating every kernel over the training set and selecting the one with highest evaluation for each feature matrix.

Logistic Regression The final classification is done by means of a logistic regression. We take the n classifier outputs as judges voting for Yes or No (written by the same author or not). For each problem, we create a vector with these votes (1 if the judge thinks the document is written by the same author and 0 if not) and use the resulting matrix to train a logistic regression model to obtain the weight of each feature category. Intuitively these weights describe the relevance of each feature category in the author's style.

To do this, we use the training data set points $x_i, i = \{1, ..., n\}$ and the set Y of the training data labels (the output of the classifiers), such that $Y = \{y : y = \{0, 1\}\}$. We want to obtain weights w_i for each point x_i solving the equation (9):

$$y = \sum_{i=1}^{n} w_i \cdot x_i \tag{9}$$

With these weights for each judge we then calculate the probability of a class in the evaluation set by the equation:

$$p(z)\frac{e^{w \cdot z}}{1 + e^{w \cdot z}} \tag{10}$$

Where z is the vector of judges for the unknown author and w is the weights vector obtained by solving (9) for $\{w_i\}$. So we take this probability to determinate if the new point is part of the given set or if it is not.

3 Results and Discussion

3.1 Results

Results of the evaluation are shown in Table 2.

Table 2. Results

Language	AUC	c@1	Combined
Dutch	0.396	0.384	0.152
English	0.517	0.5	0.258
Greek	0.589	0.56	0.33
Spanish	0.454	0.48	0.217

3.2 Discussion

Various problems exist in the methodology. First, the model that the logistic regression learns is a single one-fit-all model for all problems. This is not ideal because one of the hypotheses is that the effectiveness of each feature category is dependent on the author

we want to identify. Thus a different linear regression model should be generated for each author.

Second, the only training data we provide to each SVM classifier is the unknown texts of a single problem. Evidently, this is less than ideal, specially in cases where there is only one unkown document. In these cases we split the unknown documents into three shorter documents, which does allow the program to run the SVM algorithm but it is not sufficient information to create a general model. A better use of the distance-to-average matrix was needed in order to account for the little information each problem presented, since we opted to not use the more traditional Impostors approach.

Finally, linear regression is not necessarily the best approach for combining the classifiers. There is much literature on ensemble classifiers and the possible ways on generating the weights or scores of each one. Some examples of factors that the combining function can take into account include assigning more weight to classifiers that correctly classified hard instances, where hard refers to the fact that none or almost none of the other classifiers correctly classified them [4].

Also, we consider that for our hypothesis to be true the combining function should always find the best features of the author no matter which feature categories were used in the first place. Yet we observe that the method performs differently depending on which feature categories were included in the experiment. For example, preliminar runs where not all feature categories had been added (specifically "Multi word terms", "End of sentence word profile" and "Character n-grams"), tended to performed better (reaching a training c@1 score of 0.8 in the Spanish dataset). This is most likely due to the logistic regression not being able to handle a large number of features at least without some selection.

4 Conclusions and Future Work

Improvement of the method is needed, especially on the way the classifiers are combined. Ideally it could use a relatively large feature set without losing performance, as the Writeprints [1] method suggests, or as previous algorithms in this same task such as [3] have successfully shown.

Also, regarding the way instances are to be compared against control examples, either the Impostors approach must be adopted or else further experimentation must be done with the distance-to-average matrix in order to truly take advantage of the information it tells us about the corpus.

Acknowledgments This work is funded by the project PAPIIT-UNAM IN400312 “Análisis estilométrico para la detección de similitud textual”, as well as CONACYT CB2012/178248 “Detección y medición automática de similitud textual”

References

1. Abbasi, A., Chen, H.: Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. ACM Transactions on Information Systems (TOIS) 26(2), 7 (2008)

2. Burrows, J.: Delta: A measure of stylistic difference and a guide to likely authorship. Literary and Linguistic Computing 17(3), 267–287 (2002)
3. Khonji, M., Iraqi, Y.: A slightly-modified gi-based author-verifier with lots of features (asgalf). In: Notebook for PAN at CLEF 2014
4. Kim, H., Kim, H., Moon, H., Ahn, H.: A weight-adjusted voting algorithm for ensembles of classifiers. Journal of the Korean Statistical Society 40(4), 437–449 (2011)
5. Padró, L., Stanilovsky, E.: Freeling 3.0: Towards wider multilinguality. In: Proceedings of the Language Resources and Evaluation Conference (LREC 2012). ELRA, Istanbul, Turkey (May 2012)
6. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of the international conference on new methods in language processing. vol. 12, pp. 44–49. Citeseer (1994)
7. Schmid, H.: Improvements in part-of-speech tagging with an application to german. In: Proceedings of the ACL SIGDAT-Workshop. Citeseer (1995)
8. Stamatatos, E.: A survey of modern authorship attribution methods. Journal of the American Society for information Science and Technology 60(3), 538–556 (2009)
9. Weinberger, K.Q., Blitzer, J., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. In: Advances in neural information processing systems. pp. 1473–1480 (2005)