Author Verification using PPM with Parts of Speech Tagging

Notebook for PAN at CLEF 2014

Sarah Harvey

University of Waterloo
sharvey@cs.uwaterloo.ca

Abstract In this paper we describe a compression-based authorship verification model used in conjunction with a parts of speech tagger. We use standard language-specific parts of speech taggers on the texts in question to generate a stream of symbols representing each word, run the PPM (Prediction by Partial Matching) algorithm on the resulting stream, and use Bobicev's method of calculating and comparing cross-entropies to determine text authorship.

1 Introduction

Authorship attribution is generally described as determining the author of a particular disputed work. This is framed as a problem of authorship identification, where, given a sample of documents of known authorship, identify which author wrote the unknown text. This may also be framed as a problem of authorship verification, where we simply want to get a binary answer of whether the author in question wrote a particular text. This is one of the tasks set for PAN2014, an evaluation lab for methods relating to various aspects of stylometry[2][3][4].

PPM is a popular off-the-shelf compression method that has been explored for attribution purposes[8][1][5], but it has primarily only been explored for raw texts, and primarily only for the English language. We extract the statistical model powering PPM and use its predictive capabilities on a per-word basis as opposed to a per-character basis in previous experiments. We pre-process the texts in question to convert into POS-symbol streams, use PPM to calculate entropies, then calculate the probability of authorship based on these entropies.

2 Methodology

The basis of our method is extremely simple: run the texts in question through a parts of speech tagger and label each corresponding part with a symbol, then run the PPM algorithm on the resulting symbols, and calculate the cross-entropy between the known and unknown texts. This method is an extension of [1], in that we use the same method of calculating cross-entropies and running a pairwise t-test on the results. We change things slightly by simply preprocessing the text by running the parts of speech tagger.

2.1 Parts of speech tagging

The effectiveness of PPM on texts is purely dependent on whether or not the context size of the model can accurately capture word usage patterns. While PPM usage has been studied on English texts[9][10], it appears that its effectiveness is relatively unknown on other forms of texts. We posit that this is due to the fact that the effectiveness of PPM per-language requires context-sizing tweaks to accurately capture patterns in symbols for other languages. We decide to reduce the texts in question to symbols representing the parts of speech of the original words, and assess the effectiveness of the PPM statistical model on per-word instead of per character instances.

We used TreeTagger[7][6] for converting texts to their parts of speech equivalents, due to the fact that it had support for English, Dutch, and Spanish language tagging. Unfortunately due to time constraints we were not able to test other English speech of taggers (of which there are numerous), nor were we able to test this method for the Greek corpus.

2.2 Prediction by Partial Matching (PPM)

PPM is a compression technique that makes use of a combination of a previously seen context along with Markov chains to generate a statistical model that represents the originally seen text. Symbols on an input stream simultaneously update the model, in addition to being arithmetically coded in the resulting output stream. For the purposes of authorship attribution it is simpler to discard the arithmetic coder and any 'state' optimizations in place for compression; we are mostly only concerned with the statistical model for which we use to calculate the next character probabilties, and thus the resulting entropies, of the text in question.

The basic methdology for PPM is as follows:

1. Assume: a context structure already containing previously seen symbols in the stream
2. Read a symbol s from the stream
3. Query the context structure as to whether an n-order chain exists containing $n - 1$ previously seen symbols along with s
4. If an n-order chain does not exist...
   (a) Query the $n - 1$-order chain with $n - 2$ previously seen symbols along with s
   (b) Repeat for progressively smaller chains until an order -1 chain is reached
5. Increment the symbol counter in the context structure for n, $n - 1$, ... -1 order chains. If the symbol does not exist, initialize it in all context-matching chains that do not yet contain the symbol s.
6. Repeat for each successive symbol s

We wrote our own version of PPM in C using a hard-coded symbol context-size of 5. Code for this is available on Github[1].

[1] https://github.com/worldwise001/ppm-c/

2.3 Cross-entropy calculation

We used Bobicev's cross-entropy calculation method for comparing entropies generated by the PPM model[1]. A rundown of the algorithm is as follows:

- Concatenate all known-author texts into a text K. Let the unknown-author text be known as text U.
- Split K into k fixed n-size text chunks, and split U into u fixed n-size text chunks.
- Generate same-entropies
  - For each k:
    - Generate a PPM statistical model on $(K - k)$
    - Run the model on k and calculate the resulting entropy.
  - For each u:
    - Generate a PPM statistical model on $(U - u)$
    - Run the model on u and calculate the resulting entropy.
- Generate cross-entropies
  - Generate a PPM statistical model on U
  - For each k:
    - Run the model on k and calculate the resulting entropy.
  - Generate a PPM statistical model on K
  - For each u:
    - Run the model on u and calculate the resulting entropy.
- Compare the same-entropies with the cross-entropies using a pairwise t-test with a null hypothesis that the values are equivalent. If the null hypothesis is accepted, then we can consider that K and U are written by the same author.

Determining the appropriate values for n and the p-value for pairwise t-test required some initial experimentation. We settled on a n value of 300 (i.e. a word chunk size of 300), for most languages, with the exception of Spanish, which had a n value of 100. We also used a standard p-value of 0.05 for most languages, with the exception of Spanish, which used a p-value of 0.01.

3 Results

3.1 Initial testing

We briefly experimented with varying the cross-entropy chunk sizes, in order to determine how the value of 350 words was determined in [1], while simultaneously comparing t-test thresholds across languages. A graph showing the results on the training texts is denoted by Figure 1.

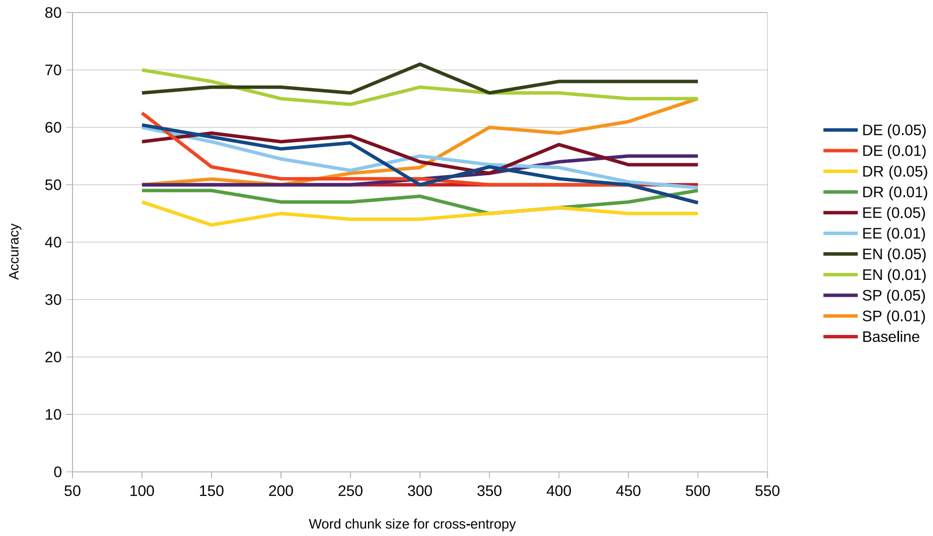

Figure 1. Initial results with varying cross-entropy chunk sizes

Table 1. Training results

Corpus	AUC	C@1	Combined
Dutch Essays	0.64497	0.61046	0.39373
Dutch Reviews	0.4391	0.5	0.2197
English Essays	0.585	0.53	0.31005
English Novels	0.6728	0.71	0.47769
Spanish Articles	0.7858	0.65	0.51077

Table 2. Test results

Corpus	AUC	C@1	Combined
Dutch Essays	0.6441	0.61458	0.39585
Dutch Reviews	0.354	0.48	0.16992
English Essays	0.57855	0.54	0.31242
English Novels	0.5398	0.525	0.2834
Spanish Articles	0.79	0.65	0.5135

4 Discussion and Future Work

We present a variant on a compression-based attribution scheme (PPM) by making the model train on parts of speech of the text, versus individual characters of the text. In this manner we can capture the style using a statistical model based on the ordering of the type of words, in contrast to the actual words being used. We found that this approach performed particularly poor for Dutch Reviews, but surprisingly well for Spanish Articles and Dutch Essays. For English Essays, performance was almost baseline, and for English Novels performance was inconsistent. We did not evaluate any Greek-language approaches in this evaluation lab.

We have a number of take-aways from this evaluation lab with regards to attributed performance and possible improvements for future work:

- It appears that an adequate choice of word chunk size is critical to the performance of this method. By reducing the chunk size for Spanish Articles to 100, we saw an improvement of 20%, that was consistent over the training and test results. Initial analysis of varying chunk size for the other languages did not show noticeable improvements, hence we simply used the size of 350 as mentioned in [1]. It is possible that more granular tuning might produce different results, as we only looked at intervals of 50 words.
- We simply used a fixed context-size of 5 for the PPM model, and did not vary this value. Not much prior work exists in experimenting with this value in authorship verification in general, so it would be interesting to see further analysis in this area.
- We only tried using a single parts of speech tagger (TreeTagger) due to the fact it had support for 3 of the languages in question in this evaluation lab. Future work would include evaluating the results of other parts of speech taggers such as the Stanford Log-linear Part-of-Speech Tagger, or the Brill Tagger. As correct tagging is imperative for our approach to work correctly, it is possible that we could see improvements in both English Essay and English Novel authorship verification if alternate English taggers are used.
- There is a clear contrast in performance between the Dutch Reviews and the Dutch Essays corpora. It is very interesting that there is a clear stylistic difference between these two corpora that this method managed to capture in such a way that the essays were more easily reidentifiable over the reviews. We posit that this is due to the fact that the essays are likely to be more clearly structured with a strong style, such that this is more readily capturable by our technique.

In general, it appears that this technique on its own does not perform exemplary in answering the binary-answer question posed by this evaluation lab: did a particular author A write a work W? It is possible that this technique may serve to augment an existing verification method by perhaps providing information regarding the ordering of type of words, versus the word usage itself. We hope to improve upon this method by coupling it with existing verification techniques that capture actual word usage.

References

1. Bobicev, V.: Authorship detection with ppm (2013)
2. Cappellato, L., Ferro, N., Halvey, M., Kraaij, W., editors: Clef 2014 labs and workshops, notebook papers. In: CEUR Workshop Proceedings (CEUR-WS.org). 5th International Conference of the CLEF Initiative (CLEF 14), Springer (2014)

3. Gollub, T., Potthast, M., Beyer, A., Busse, M., Rangel, F., Rosso, P., Stamatatos, E., Stein, B.: Information access evaluation meets multilinguality, multimodality, and visualization. In: Recent Trends in Digital Text Forensics and its Evaluation. 4th International Conference of the CLEF Initiative (CLEF 13), Springer (2013)
4. Juola, P., Stamatatos, E.: Overview of the author identification task at pan 2013. In: CLEF 2013 Evaluation Labs and Workshop–Working Notes Papers (2013)
5. Khmelev, D.V., Teahan, W.J.: A repetition based measure for verification of text collections and for text categorization. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval. pp. 104–110. SIGIR '03, ACM, Toronto, Canada (2003), http://doi.acm.org/10.1145/860435.860456
6. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of International Conference on New Methods in Language Processing (1994)
7. Schmid, H.: Improvements in part-of-speech tagging with an application to german. In: Proceedings of the ACL SIGDAT-Workshop (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. Teahan, W.J., Harper, D.J.: Using compression-based language models for text categorization. In: Language Modeling for Information Retrieval, pp. 141–165. Springer (2003)
10. Teahan, W.J.: Text classification and segmentation using minimum cross-entropy. In: RIAO. pp. 943–961 (2000)