Using Grammatical Markov Models for Stylometric Analysis

CS224N : Natural Language Processing

Erik Goldman
erik@stanford.edu

Abel Allison
aallison@stanford.edu

Abstract

Researchers have tackled the problem of authorship attribution in several different ways, using various metrics to identify the author of an anonymous document given a set of writing samples from potential candidates. Common complaints about modern methodologies tend to accuse studies of content bias, which occurs when quantitative models identify similar content rather than similar styles. This artificially increases accuracy by producing good results on test data while failing to identify authors in real-world applications. We examine several quantitative methods that isolate style by using grammar-based features rather than relying on models of word shape and frequency.

1 Introduction

The ability to determine the author of an anonymous text has many applications, ranging from historical scholarship to modern forensic work. Most methods of author attribution have taken a quantitative approach, comparing statistical features of a writing sample against those of potential authors. Authorship attribution algorithms typically start with a learning phase during which a learning algorithm processes several texts by potential authors to derive some stylistic metrics. This approach has several pitfalls that must be carefully avoided. Authors typically write with different styles depending on a work's intended audience, for example. Authors that publish in multiple journals, magazines, or newspapers may have their writing style tainted by aggressive editors. These issues add strong confounding variables in stylometric analysis and make training difficult.

We examined the algorithmic problem of ensuring that features isolate the writing style from the writing content. Content words are tokens that vary based only on the subject about which the author is writing– words that are indicative of topic rather than style. The more that features rely on content, the less reliable they are, as they will increasingly report false positives for works on the same subject and false negatives for works with different content. If an algorithm uses word frequency as a metric, for example, it may pick up on locations and proper nouns related to a specific topic. Such an algorithm will clearly favor any document that contains these locations and names, regardless of that document's author.

We attempted to eliminate such content-based problems by investigating metrics that discard word data entirely, relying only on grammatical features to build authorship models. Using a probabilistic parser, we extracted a full syntax tree from each sentence and used this data to perform our analysis. We created two features using this data, both based on bi-gram Markov models. These features have no relation to the content of the document, and only relate to its grammatical structure, providing a guarantee that content would not confound our analysis.

For comparison, we also included two metrics that work solely on word shape and frequency in our analysis.

2 Related Work

Many researchers have done similar work examining and evaluating various metrics. Diederich et al. (2003) apply Support Vector Machine methods to the problem in hopes that it copes with the large amount of inputs without requiring the definition and weighting of different features. They saw decent success (60-80%), however, their accuracy goes down when they use just grammatical features and word shape.

Visa et al. (2001) use document prototypes, interesting documents or part of extracted, salient texts, to match with a document database. Usually applied to extracting document meaning, the authors use the technique for authorship attribution.

Burrel and Rousseau (1995) use a numerical method of fractional counts to make certain assumptions about the distribution of authors and works written which improves their ability to do author attribution. Using these assumptions they are able to make more concrete observations about fractional counts graphs in cases of one or multiple authors.

3 Algorithms and Methods

3.1 Data

We chose to use fiction as our dataset since fiction writers are often characterized by their styles as opposed to journalists, who print a much more literal account of events. To make the problem more interesting, we chose to restrict our corpus to a single genre: detective novels from the 1920's. The works we used came from the *Project Gutenberg* website, a collection of old texts whose copyrights have expired. We currently are testing on five different authors: E. W. Hornung, Maurice Leblanc, Sax Rohmer, J. S. Fletcher, and Arthur J. Rees. We chose only those authors which had enough work to provide a variety of contexts and styles, and who did not exclusively use recurring characters and locations. All were prominent mystery novelists in the late 19th and early 20th centuries. For each author, we have at least three works: two training novels and one test novel.

3.2 Features

The following features were investigated and evaluated:

1. **"Limit" Word Frequency**

   This feature counts words that occur frequently across multiple works by the same author. The purpose of this is to find certain "function" words that the author frequently uses in his writing that transcend particular styles he/she might employ towards a particular audience or content. This was included to compare our grammar-based and word-based metrics side-by-side.

2. **Grapheme Frequency**

   This feature looks at histograms of grapheme counts, where a grapheme is any alphanumeric or punctuation character. The counts are made and then normalized with respect to each other. This is another word-based metric used for comparison.

3. **Part-of-Speech Bigram Model**

   This feature looks at a sentence's grammatical parse from the top-down. We parse the sentence to get a hierarchy of tags, and then calculate the probability

   $$P(children|parent) = \frac{C(parent \rightarrow children)}{C(parent)}$$

   over all of an author's documents.

4. **Preterminal Tag Bigram Model**

   This feature looks at a sentence's grammatical parse from left to right. It examines all preterminal (or part-of-speech) tags and calculates the probability

   $$P(A|B) = \frac{C(t_{i-1} = B \wedge t_i = A)}{C(B)}$$

   , where t_i is the tag for word i in the sentence.

3.3 Comparison Metrics

Once we have gathered statistics, we must compare our data so that we can determine how similar an unknown and known author are. To do so, we used a voting system in which each feature examines a text and a set of potential authors and chooses the writer with the best feature score. Each feature is hand-weighted according to its observed effectiveness. A method such as Expectation Maximization could be used to learn weights, however we decided this was unnecessary since our data set is so small and we risk over-fitting.

Once the histograms are compiled, you need to be able to compare them. We used two methods to do so. We wanted to see how results differed when using differing comparison metrics.

1. **Chi-Squared Metric [5]**

   Each feature is simply a histogram of counts. The most commonly used metric for comparing two histograms is the Chi Squared method, which computes:

   $$\chi^2 = \sum((O_i - E_i)^2/E_i) \; i = 1,2,3,\ldots,n$$

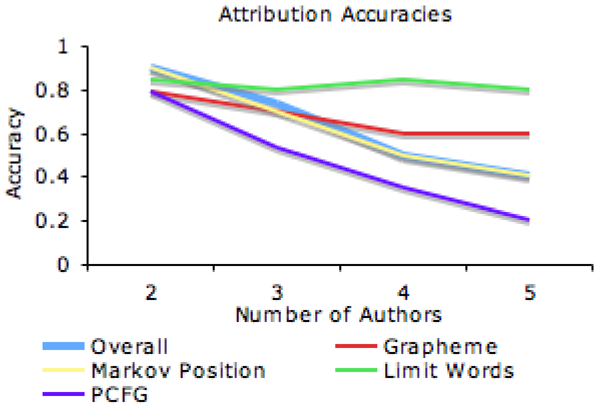
   where O is the set of observed frequencies (the unknown work) and E is the set of expected frequencies (our training data). A low chi-square value corresponds to a closer histogram match. Once the chi-square values are computed, we rank authors in ascending order and take the lowest one as the vote for that feature.

2. **Difference Formula [6]** The similarity formula is similar to the basic Chi-squared test, however with larger data sets, the n-gram frequencies tend to vary much more. The result is that more frequent n-grams are emphasized more. The following similarity formula is used to account for this by normalizing the differences by dividing them by the average frequency for the given n-gram. This results in the following formula:

   $$\sum_{n \in \text{profile}} \left(\frac{2 \times (f_1(n) - f_2(n))}{f_1(n) + f_2(n)} \right)^2$$

4 Results

To collect data, we ran our model on the power set of all our authors. For each element of the power set, we used each author in that element as an unknown. The result was 75 tests. Results are shown below, where accuracies are measure by number correct divided by number tested.



Percentage accuracy for various metrics;

# Authors:	2	3	4	5
Combined	0.895	0.733	0.5	0.4
Grapheme	0.790	0.7	0.6	0.6
Markov	0.895	0.7	0.5	0.4
LimWord	0.842	0.8	0.842	0.8
PCFG	0.790	0.533	0.35	0.2

5 Analysis of Errors: Combined Metrics

Our final, combined algorithm did not scale particulary well to multiple authors. We blame this on several factors.

First, the voting system that we used was extremely imprecise. Each feature contributes a binary vote, choosing a single candidate and giving him or her the full weight assigned to the metric. This became extremely problematic in situations like on this analysis of a J. S. Fletcher novel, which was mischaracterized as being a Maurice LeBlanc work:

```
grapheme profile
Name: J. S. FLETCHER score: 0.004997694575545716
Name: E. W. HORNUNG score: 0.01648918203997684
Name: MAURICE LEBLANC score: 0.014157823293436162
Name: ARTHUR J. REES score: 0.01785185233280465
Name: SAX ROHMER score: 0.012964794013682335
grapheme profile votes for J. S. FLETCHER
2-gram profile
markov pos
Name: J. S. FLETCHER score: 1126.6965679365971
Name: E. W. HORNUNG score: 1324.8749512880452
Name: MAURICE LEBLANC score: 1056.8361163886955
Name: ARTHUR J. REES score: 1218.3830562064386
Name: SAX ROHMER score: 1280.9943042571722
markov pos votes for MAURICE LEBLANC
```

In this example, the grapheme profile declared J.S. Fletcher a clear winner by a factor of 3, whereas the part-of-speech Markov model was less certain, and actually ranked J. S. Fletcher as its runner up. By making each feature more continuous, we could more properly blend feature results together, ensuring that confident features received more weight than a mostly homogenous set of scores. We might also consider giving some weight to second- or third-place finishers, rather than simply selecting a single winner and giving him or her all of the feature's weight.

In short, we do not believe that our features were flawed– rather, our technique of combining them simply did not allow them to fully utilize all of the information that they provided to us. In general, when features scored a single candidate significantly differently than the pack, that candidate tended to be correct. A combination of learning weights through the EM algorithm and a more continuous scaling system would enhance our results greatly.

6 Performance Considerations

A downside of grammatical features is the time taken to parse a sentence. Each sentence takes a couple of seconds to parse, and our corpus of novels consists of thousands of sentences. Our average parse time was around 45 minutes per novel, which is clearly unacceptable for the quality of results that we achieved.

The redeeming quality of this bottleneck is that the ”database” of statistics only needs to be compiled once. Each novel only has to be parsed a single time, after which we can serialize the raw data so that it can be reloaded and analyzed at a later time. Still, the necessity of using a probabilistic parser makes rapid iteration a difficult task and presents a stumbling block for researchers who need to see the effects of their changes quickly to know whether they're on the right direction. In addition, it ties the accuracy of the feature to the accuracy of the parser (although in practice, parsing is very accurate).

7 Limitations

The fact that we limit our texts to a particular genre is both good and bad. It is good in that it decreases the significance of many ”function” words that might be common to a certain genre. Since all the works come from detective novels, the words 'gun' or 'solved' will probably be common to all works. If we had mixed genres, it would be very easy to distinguish a romance from another genre just by looking at words that might only happen in a single genre. In this way, we made the problem more interesting and compelling.

On the other hand, limiting our texts to a certain genre really only allows us to provide an intrinsic measure of our success, where it would be much more attractive to show extrinsic indicators. Since all of our works come from detective novels, many content-based statistics will be similar across authors, varying only in function word usage. This allows even the strictest content-based metrics to perform fairly accurate stylometric analysis, despite their unsuitability for the task. The ideal scenario for us would be to have novels of multiple genres written by the same author, which would allow us to separate an author's style from the subject of his or her writing. Unfortunately, few authors write for multiple genres, and even fewer have novels available in the public domain.

The other strong limitation to our results is the number of texts per author used. Ideally, we would have many texts compiled into our database. The slow run-time of the parser combined with the need to recompile our database for development purposes made a small corpus size a necessity. We thus capped the number of authors at 5, and only analyzed 3 works from each one.

8 Conclusion

Suprisingly, our grammatical metrics did not perform as well as we expected. Despite their ability to remove content entirely from the authorship attribution process, they did not prove to be useful metrics worth investigating further. While they would almost always select the correct author in the top-two candidates that they returned, they exhibit significantly worse performance than the simple metrics that we put in for comparison. While it's easy to blame this on content bias, all of our test data was on the same subject and there was no plot correlation between an author's individual novels.

It appears that word shape and frequency is a much better indicator of style than simple grammatical derivations. The common wisdom about so-called ”function” words being the best indicator of style may be true; function words are defined as content-invarient words and phrases common to a specific author. Analysis of these words and phrases may be sufficient for authorship attribution, whereas grammar-only features seem to have poor accuracy when scaling to even 5 candidates.

One can achieve function-word isolation in several ways, but using a ”word-limit” profile seems to be the most effective. In this scheme, one only adds a word and its associated frequency to an author's dataset if that word appears in at least *n* documents by that author, where *n* varies depending on context. For an author with many similar works, *n* will be larger, whereas intersecting even two or three of a journalist's articles may produce a perfect set of function word frequencies.

Based on our results, we believe that the majority of authorship analysis should focus more on function word shape and frequency, and less on grammatical analysis.

9 References

[1] **Aizawa, A.** (2001) Linguistic Techniques to Improve Performance of Automatic Text Categorization. *Proceedings 6th NLP Pac. Rim Symp.*

[2] **Burrell, Q. and Rousseau, R.** (1995). Fractional Counts for Authorship Attribution: A Numerical Study. *Journal of the American Society for Information Science*. 46: 97-102.

[3] **Chaski, C.E.** (2005). Who's At the Keyboard? Authorship Attribution in Digital Evidence Investigations. *International Journal of Digital Evidence.* 4:np.

[4] **Diederich, J., Kindermann, J., Leopold, E., and Paass, G.** (2003). Authorship Attribution with Support Vector Machines. *Applied Intelligence*. Dordrecht: Springer Netherlands, 19: 109-123.

[5] **Grieve, J.** (2007). Quantitative Authorship Attribution: An Evaluation of Techniques. *Literary and Linguistic Computing Advance Access*.

[6] **Keselj, V., Peng, F., Cercone, N., and Thomas, C.** (2003). N-Gram-Based Author Profiles for Authorship Attribution. *Proceedings of the Conference Pacific*. Halifax: Dalhousie University.

[7] **Malyutov, M.B.** (2005). Authorship Attribution of Texts: A Review. *Electronic Notes in Discrete Mathematics*. Boston: Elsevier B.V., 21: 353-357.

[8] **Stamatatos, E., Fakotakis, N., Kokkinakis, G.** (2001). Computer-based Authorship Attribution Without Lexical Measures. *Computers and the Humanities*. Netherlands: Kluwer Academic.

[9] **Visa, A., Autio, S., Makinen, J., Back, B., and Vanharanta, H.** (2001). Data Mining of Text as a Tool in Authorship Attribution. *SPIE proceedings series*.