A Classifier System for Author Recognition Using Synonym-Based Features

Jonathan H. Clark, Charles J. Hannon

Department of Computer Science, Texas Christian University
Fort Worth, Texas 76129
{j.h.clark, c.hannon}@tcu.edu

Abstract. The writing style of an author is a phenomenon that computer scientists and stylometrists have modeled in the past with some success. However, due to the complexity and variability of writing styles, simple models often break down when faced with real world data. Thus, current trends in stylometry often employ hundreds of features in building classifier systems. In this paper, we present a novel set of synonym-based features for author recognition. We outline a basic model of how synonyms relate to an author's identify and then build an additional two models refined to meet real world needs. Experiments show strong correlation between the presented metric and the writing style of four authors with the second of the three models outperforming the others. As modern stylometric classifier systems demand increasingly larger feature sets, this new set of synonym-based features will serve to fill this ever-increasing need.

"The least of things with a meaning is worth more in life
than the greatest of things without it."
Carl Jung (1875 - 1961)

1 Introduction

The field of stylometry has long sought effective methods by which to model the uniqueness of writing styles. Good models have the quality that they can differentiate between the works of two different authors and label them as such. However, even some of the best models suffer from deficiencies when presented with real world data. This stems from the fact that a writing style is a very complex phenomenon, which can vary both within a literary work and over time. [12] Given these challenges, it is not surprising that the field of stylometry has not yet discovered any single measure that definitely captures all the idiosyncrasies of an author's writings.

Recently, the field of stylometry has moved away from the pursuit of a single "better" metric; modern computational approaches to author recognition combine the power of many features. [11, 14] Thus, the field has begun to recognize that the problem of author recognition is much like a puzzle, requiring the composition of many pieces before the picture becomes clear. In this paper, we present a novel set of

synonym-based features, which serves as yet a few more pieces of the much larger puzzle.

Why do we propose a feature set based on synonyms? By examining words in relation to their synonyms, we concern ourselves with the meaning behind those words. For the proposed features, we are primarily interested in answering the question "What alternatives did the author have in encoding a given concept in this language?" In answering this question, we find that we obtain a metric which has a strong correlation with writing style.

1.1 Task

The most common application of the techniques discussed in this paper will likely be within a classifier system for author identification. For this task, we are given a set of known authors and samples of literature that are known to correspond to each author. We are then presented with a text sample of unknown authorship and are asked "Of the authors that are known, who is most likely to have written this work?"

1.2 Related Work

Some of the earliest features used for author recognition include word length, [1, 4] syllables per word, [3] and sentence length. [8] Though these measures are found to be insufficient for the case of real world data by Rudman, [11] they did make progress in the computational modeling of an author's writing style. These methods became somewhat more sophisticated with the study of the distinct words in a text by Holmes. [6] Stamatatos et al. present a method that utilizes a vector of 22 features including both syntactic and keyword measures. [13] More recent efforts have gone below the level of the lexicon and examined text at the character-level. [7, 10]

The relation of writing style and synonyms is an area that has been much less studied. Coh-metrix, a tool for text analysis based on cohesion calculates measures as polysemy (words having more than one meaning) and hypernymy (words whose meaning is on the same topic but has a broader meaning). [5] However, these measures were not used for determining what alternative representations of a concept an author had to choose from as is the case in the presented work.

This paper builds on the work of Clark and Hannon. [2] However, this previous work targeted flexibility over accuracy and was evaluated on non-contemporary authors. In this paper, we begin by refining the previous work into a new theoretical framework suitable for combination with other feature sets and present it as model 1. We then present enhancements that cope with the shortcomings of model 1 and compare all 3 models using a more difficult data set.

2 Theory

The goal in developing a good model of an author's writing style is to capture the idiosyncratic features of that author's work and then leverage these features to match a work of unknown authorship to the identity of its author. As previously stated, a modern system can use hundreds of features at a time. However, each of these features must have a significant correlation with some component of writing style that varies between authors.

We propose that an author's repeated choice between synonyms represents a feature that correlates with the writing style of an author. Not only do we want to measure which words were selected, but how much choice was really involved in the selection process. For instance, given the concept of "red," an author has many choices to make in the English language with regard to exactly which word to select. The language provides many alternatives such as "scarlet" with which an author can show creative expression. More importantly, this creative freedom leads authors to make unique decisions, which can later be used as identifying features. Contrast the example of colors with the word "computer." It is a concept that maps to relatively few words. Therefore, we might say that an author had less opportunity for expression and that this word is less indicative of authorship.

In the following sections, we present three models, which each represent a point in the natural evolution of this work. Model 1 captures the basic concept of how synonyms relate to an author's identity while ignoring some of the subtleties of the underlying problem. However, it serves as a conceptual springboard into the more refined models 2 and 3, which perform a deeper analysis of each word to obtain better performance on real world data.

2.1 Model 1

Model 1 demonstrates at the most basic level how synonyms can be tied to an author's identity. Loosely speaking, the idea behind model 1 is that if a word has more synonyms, then the author had more words from which to choose when encoding a given concept. Therefore, the word should be given more weight since it indicates a higher degree of free choice on the part of the author. We model this concept in terms of our task of identification of an unknown author by collecting a feature vector for each word in an author's vocabulary, running an algorithm over the feature vector, and finding the argument (author) that maximizes the function's value.

We define the feature vector f_1 of a word w as having the following elements[1]:

- The number of synonyms s for w as according to the WordNet lexical database [9]
- The shared text frequency n for w; that is, if author a uses word w_a with frequency n_a and author b uses word w_b with frequency n_b then the shared frequency $n = \min(n_a, n_b)$.

[1] For clarity, variables peculiar to model 1 are given a subscript of 1.

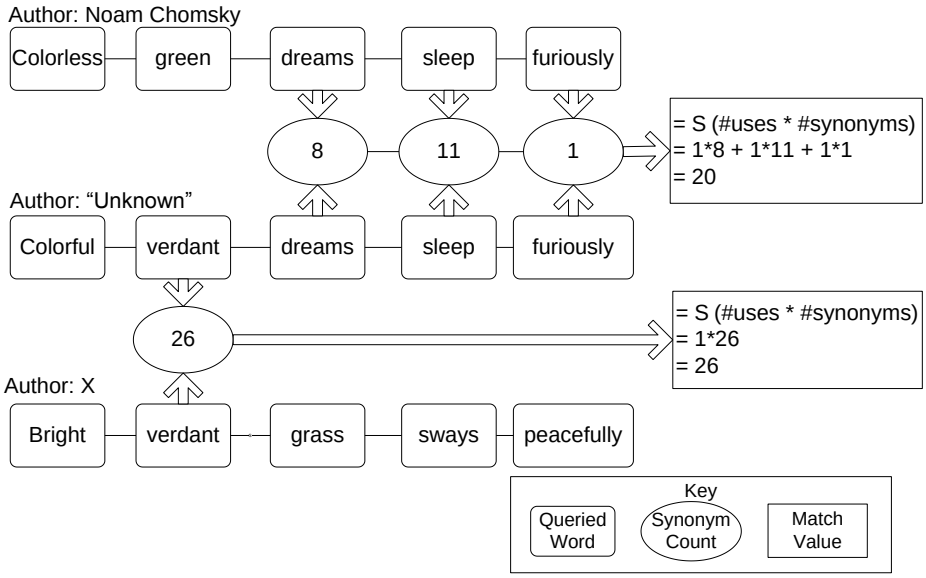

Fig. 1. An example of how match values are calculated for model 1. The top and bottom sentences represent training samples for the authors Noam Chomsky and a hypothetical Author X, respectively. The middle sentence represents an input from an author whose identity is hidden from us. We then perform calculations as shown to determine the author's identity

Next we define the function $match_1$, which generates an integer value directly related to the stylistic similarity of the unknown author u with the known author k:

```
function match1(u,k)
  m ← 0
  for each unique word wu used by author u
    for each unique word wk used by author k
      if wu = wk then
        generate f1 of wu,wk
        m ← m + f1[n] * f1[s]          (see definition of f1 above)
      end if
    end for
  end for
  return m
end function match1
```

Finally, we define our classifier such that the identity I of the unknown author is

$$\mathrm{I} = \arg\max_{k \in T} \; match_1(u,k) \qquad (1)$$

where T is the set of all known authors on which the system was trained.

As a concrete example, consider the above example. (Fig. 1) The words "dreams," "sleep," and "furiously" have 8, 11, and 1 synonym, respectively while the word "verdant" has 26 synonyms. A traditional bag-of-words approach would select Noam Chomsky as the author since the sentence of unknown authorship has 3 word

matches with Noam Chomsky's vocabulary. However, model 1 takes into account the fact that the word "verdant" has 26 synonyms and gives it more weight than that of all of the other words in the figure. Thus, model 1 selects Author X as the author of the unknown sentence. Having set forth a simplified model, we now turn to the matter of designing a model robust enough to deal with real world data.

2.2 Model 2

In building model 2, we sought to eliminate some of the issues that presented themselves in the implementation and testing of model 1. A careful analysis of the output of model 1 demonstrated two key weaknesses:

1. A handful of the same high frequency words including pronouns and helping verbs (e.g. "it" or "having") were consistently the largest contributors to the value returned by the *match* function even though to a human observer, they are clearly not unique markers of writing style
2. Each synonym was being treated as equal although logic suggests that a more common word such as "red" is not as important as an infrequent word such as "scarlet" in determining the identity of an author

To handle the first case in which high frequency words were masking the effect of lower frequency words, we added two improvements over model 1. First, we define a global stopword list that will be ignored in all calculations, a common practice in the field of information retrieval. This reduced the amount of noise being fed to the classifier in the form of words that have lost their value as identifying traits. Second, we revise the function *match* such that we divide the weight for a matched word by the global frequency of that word. The global frequency is computed either via the concatenation of all training data (as is the case for the presented experiments) or via the some large corpus.

In response to the second issue, we see that it is desirable to give words different weights depending on their text frequency. Recall that we seek not only to consider what word choices the author made, but also to consider what the author's alternative choices were in encoding this concept. Thus, we do not only include the text frequency of the word, but the sum over the global frequencies of all synonyms of each word the author chooses (shown in the example on the following page). Seen in a different light, we sum the frequencies of all words an author could have chosen for a given concept. In this way, we obtain a value that not only corresponds to the number of choices the author had, but also how idiomatic those choices are with regard to common language usage.

To summarize, we define the model 2 feature vector f_2 of a word w as having all elements of f_1 with the following additional elements:

- Whether or not w is contained in the stop list
- The global frequency g of w
- The sum u over the global frequencies of all synonyms of w

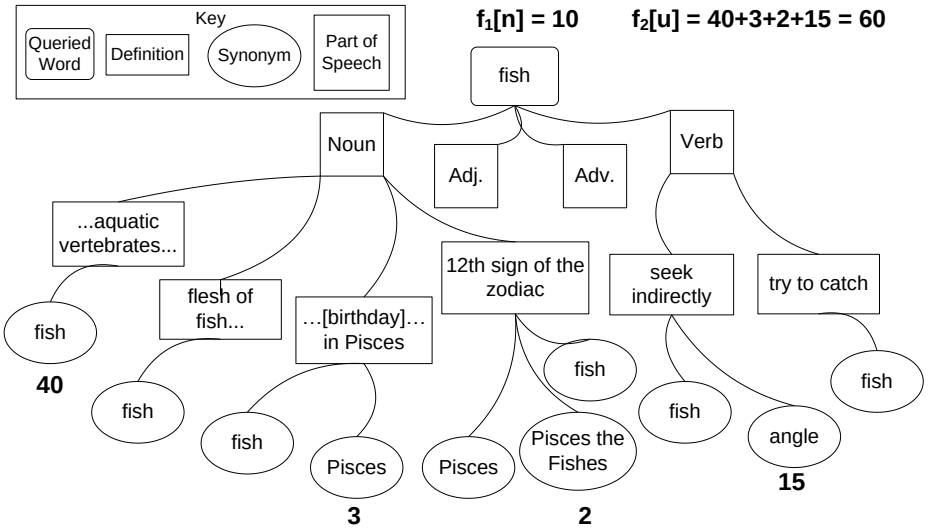

Fig. 2. An example of a word *(fish)* and its synonyms using the hierarchy defined by WordNet. For sake of discussion, arbitrary weights have been placed under the returned synonyms. These are used to provide context for subsequent examples of models 2 and 3

The modified version of the function *match*, which we will refer to as $match_2$, now generates a real value (as opposed to integer) and behaves as follows:

```
function match2(u,k)
  m ← 0.0
  for each unique word wu used by author u
    for each unique word wk used by author k
      if wu = wk AND wu,wk is not in stoplist then
        generate f2 of wu,wk
        m ← m + f1[n] * f2[u] / f2[g]   (see definition of f2 above)
      end if
    end for
  end for
  return m
end function match2
```

To again give a more tangible example of how the model works, we present Fig. 2. Assume that the vocabularies of both the unknown author *u* and the known author *k* contain the word "fish" and that they used the word 10 and 15 times, respectively. Thus, the word has a shared frequency $f_1[n]$ of 10. Further, assume that "fish" occurred 20 times in some large corpus from which we obtain the global frequency. Since "fish" is not a stop word, it will be given a non-zero weight. Also note that fish has four unique synonyms with global frequencies of 40, 3, 2, and 15, respectively. Thus, the sum over the global frequencies of the synonyms *u* is 60. With this information we can now calculate the value of *m* as shown in the function $match_2$ by 10 * 60 / 20 = 30.

The additional features in model 2 make it much more robust than model 1. It considers not only the number of alternative choices an author had, but how idiomatic those choices are with regard to how language is commonly used. We now look toward model 3, which attempts to incorporate still more linguistic information into the synonym-based feature set.

2.3 Model 3

In model 3, we attempt to exploit the morphology of the English language. Though English is not considered a morphologically rich language, it certainly does have cases in which the morphology causes what the average speaker might consider the same word to be mapped to two different words (e.g. "give" and "gives").

Model 3 attempts to compensate for this phenomenon by applying stemming to each word in the author's vocabulary. This process of stemming is the only change between models 2 and 3. The assumption here is that it is not important which morphological form of a word an author chooses. Rather, in model 3, we place the emphasis on which synonym and which shade of meaning an author chooses to represent a given concept. We leave it up to the results to indicate whether or not this is a meaningful assumption.

3 Implementation

3.1 Corpus

To perform the author identification task, we selected a corpus consisting of 1,333,355 words from four authors including Jacob Abbott, Lydia Child, Catharine Traill, and Charles Upham. To ensure our system was not using stylistic markers of time periods in differentiating between authors, the authors were selected such that they were all born within roughly a year of each other (1802 – 1803). All works used in the test set were retrieved from Project Gutenberg[2] and are freely available for download. After obtaining the data, we removed all portions of the text that would not be considered an author's original work (i.e. tables of contents, prefaces, etc.). The remaining body of text was then divided evenly into five folds, one of which was used as training data and the other four being left as test cases. Basic statistics for the corpus are presented in Table 1.

[2] Project Gutenberg is accessible at http://www.gutenberg.org.

Table 1. This table shows word counts for each fold of the 1,333,355 word corpus.

	Total Words		Unique Words	
Author	Testing (Avg)	Training	Testing (Avg)	Training
Abbott	60,316	57,898	4,763	6,198
Child	87,187	90,960	7,646	6,963
Traill	59,713	63,482	6,576	7,168
Upham	57,987	57,075	6,297	6,858

3.2 WordNet

One very important tool in implementing the system was Princeton WordNet.[3] [9] WordNet is a lexical database of the English language that has qualities similar to both a dictionary and a thesaurus. Most importantly, it contains links between synonyms which may be traversed as "synsets." For example, Fig. 2 shows a synset taken from WordNet. Version 2.1 of WordNet, used in this research, contains 207,016 word-sense pairs within 117,597 synsets. WordNet also includes a very simple yet effective morphological processor called Morphy, which we used to perform stemming for model 3.

3.3 Stop Word List

To prevent conflict of interest, we used a stop word list from an external source, the Glasgow University Information Retrieval group[4]. The list contained 319 of the most common words in the English language. At runtime, we used the WordNet Morphy morphological processor to stem the words on the Glasgow stop list to obtain more stop words. Finally, we augmented this list with names from the U.S. Census Bureau website, which included the most frequent 90% of both first and last names, as indicated by the 1990 census. [5] The combination of all these sources was used as the stop word list for models 2 and 3.

3.4 Pre-Processing

Incident to using WordNet, part of speech tagging is recommended so that WordNet can narrow down which senses of the word might be intended (see Fig. 2). For this purpose, we employed the Stanford Log-Linear Part of Speech Tagger.[6] [15] The

[3] The can be downloaded at http://wordnet.princeton.edu/
[4] This stop word list is located at http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/
[5] The name list is available at http://www.census.gov/genealogy/www/freqnames.html
[6] The tagger may be obtained at http://nlp.stanford.edu/software/tagger.shtml

supplied trained tagger was used as there was no compelling reason for custom training.

4 Results

Results for each section are presented for the three cases of classifying between 2, 3, or 4 authors at a time. For all cases, all 4 test folds of each author were evaluated against some number of trained models. In the case of classifying between 3 authors at a time, all possible ${}_4C_3$ (4) combinations of 3 authors were evaluated and results were then averaged over these sets. Similarly, for the case of classifying between 2 authors at a time, all ${}_4C_2$ (6) combinations were tested. Results are reported as precision, recall, and F1 scores. Precision is defined as the number of test cases (i.e. folds) correctly reported as being written by a given author divided by the total number of test cases reported as being written by that author. Similarly, recall is defined as the number of test cases correctly reported divided by the total number of correct test cases possible. Finally, the F1 score is calculated as the harmonic mean of precision and recall.

4.1 Model 1

We begin by analyzing the performance of model 1. Of the three models, model 1 was produced the lower overall F1 scores (see Table 2). For the case of differentiating between two authors at a time, model 1 produced better than chance results. As model 1 has to deal with choosing between more authors, performance declines steeply. Certainly we prefer a model that displays both higher accuracy and more graceful degradation when faced with larger numbers of authors. To realize these characteristics, we turn to models 2 and 3.

Table 2. Precision, recall, and F1 scores for model 1.

	Authors = 4			Authors = 3			Authors = 2		
Author	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Abbott	0.000	0.000	0.000	0.000	0.000	0.000	0.333	1.000	0.500
Child	1.000	0.267	0.421	1.000	0.353	0.522	1.000	0.522	0.686
Traill	0.250	1.000	0.400	0.500	0.462	0.480	0.750	0.563	0.643
Upham	0.000	0.000	0.000	0.083	1.000	0.154	0.417	1.000	0.588
Overall	0.313	0.313	**0.313**	0.396	0.396	**0.396**	0.625	0.625	**0.625**

4.2 Model 2

Model 2 exhibited the most desirable qualities of all the models evaluated. Not only was it highly accurate in terms of F1 score, but it also displayed a graceful degradation curve as it was faced with discerning between larger numbers of authors. The benefits of having probed more deeply into the frequency of all of a word's synonyms and utilizing global frequencies in our feature vector are underlined by these results (see Table 3).

Table 3. Precision, recall, and F1 scores for models 2 and 3.

	Authors = 4			Authors = 3			Authors = 2		
Author	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Abbott	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Child	1.000	0.080	0.889	1.000	0.857	0.923	1.000	0.923	0.960
Traill	0.750	1.000	0.857	0.833	1.000	0.909	0.917	1.000	0.957
Upham	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Overall	0.938	0.938	**0.938**	0.958	0.958	**0.958**	0.979	0.979	**0.979**

4.3 Model 3

Having performed the additional step of stemming for model 3, the expected result was that scores would increase. In actuality, there was no change from the scores of model 2 (Table 3). To clarify the meaning of these results, we also calculated the average percent difference between the weights returned by the *match* function for the top two authors (Table 4). This gives us a rough estimate of how "confident" the system was in making its choice with a larger percentage difference being more desirable. For all cases, model 2 produced these larger differences between its top 2 matches. Thus, we conclude not only that we received no benefit from stemming, but that it had a negative effect on the output, be it very small negative effect. From this, we draw that the author's choice about which form of a word to use is an important choice and should not be discarded via stemming.

Table 4. This table shows the percent difference between the weights returned by the *match* function for the top two authors, averaged over all test cases.

	Authors = 4		Authors = 3		Authors = 2	
Author	Model 2	Model 3	Model 2	Model 3	Model 2	Model 3
Abbott	0.136	0.051	0.137	0.077	0.157	0.125
Child	0.120	0.160	0.150	0.188	0.204	0.249
Traill	0.146	0.104	0.168	0.125	0.218	0.179
Upham	0.144	0.061	0.196	0.083	0.265	0.127
Overall	**0.135**	0.098	**0.164**	0.121	**0.211**	0.172

5 Conclusion

We have presented a novel set of synonym-based features for use in a classifier system that performs author identification. As evidenced in the results, these features perform well on real world data when properly tuned (i.e. models 2 and 3). However, to harness the full potential of this feature set, it should be combined with many other features so that a full range of characteristics of writing style are considered. This new set of synonym-based features provides yet another tool with which stylometric classifier systems will be able to analyze written language.

References

1. Brinegar, C.S.: Mark Twain and the Quintus Curtius Snodgrass Letters: A Statistical Test of Authorship. Journal of the American Statistical Association 58 (1963)
2. Clark, J.H., Hannon, C.J.: An Algorithm for Identifying Authors Using Synonyms. ENC 2007 (2007)
3. Fucks, W.: On the mathematical analysis of style. Biometrika 39 (1952) 122-129
4. Glover, A., Hirst, G. (eds.): Detecting stylistic inconsistencies in collaborative writing. Springer-Verlag, London (1996)
5. Graesser, A.C., McNamara, D.S., Louwerse, M.M., Cai, Z.: Coh-Metrix: Analysis of text on cohesion and language. Behavior Research Methods, Instruments, and Computers 36 (2004) 193-202
6. Holmes, D.I.: Authorship attribution. Computers and the Humanities 28 (1994)
7. Khmelev, D.V., Tweedie, F.J.: Using Markov Chains for Identification of Writers. Literary and Linguistic Computing 16 (2002) 299-307
8. Mannion, D., Dixon, P.: Sentence-length and Authorship Attribution: the Case of Oliver Goldsmith. Literary and Linguistic Computing 19 (2004) 497-508
9. Miller, G.A.: WordNet: A Lexical Database for English. Communications of the ACM 38 (1995) 39-41
10. Peng, F., Schuurmans, D., Keselj, V., Wang, S.: Language Independent Authorship Attribution using Character Level Language Models. 11th Conference of the European Chapter of the Association for Computational Linguistics (2004)
11. Rudman, J.: The State of Authorship Attribution Studies: Some Problems and Solutions. Computers and the Humanities 31 (1998) 351-365
12. Smith, J.A., Kelly, C.: Stylistic constancy and change across literary corpora: Using measures of lexical richness to date works. Computers and the Humanities 36 (2002) 411-430
13. Stamatatos, E., Fakotakis, N., Kokkinakis, G.: Automatic Text Categorization in Terms of Genre and Author. Computational Linguistics 26 (2000) 471-495
14. Stamatatos, E., Fakotakis, N., Kokkinakis, G.: Computer-Based Authorship Attribution Without Lexical Measures. Computers and the Humanities 35 (2001)
15. Toutanova, K., Klein, D., Manning, C., Singer, Y.: Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. HLT-NAACL (2003) 252-259