Literary authorship attribution with phrase-structure fragments

Andreas van Cranenburgh
Huygens ING
Royal Netherlands Academy of Arts and Sciences
P.O. box 90754, 2509 LT The Hague, the Netherlands
andreas.van.cranenburgh@huygens.knaw.nl

Abstract

We present a method of authorship attribution and stylometry that exploits hierarchical information in phrase-structures. Contrary to much previous work in stylometry, we focus on content words rather than function words. Texts are parsed to obtain phrase-structures, and compared with texts to be analyzed. An efficient tree kernel method identifies common tree fragments among data of known authors and unknown texts. These fragments are then used to identify authors and characterize their styles. Our experiments show that the structural information from fragments provides complementary information to the baseline trigram model.

1 Introduction

The task of authorship attribution (for an overview cf. Stamatatos, 2009) is typically performed with superficial features of texts such as sentence length, word frequencies, and use of punctuation & vocabulary. While such methods attain high accuracies (e.g., Grieve, 2007), the models make purely statistical decisions that are difficult to interpret. To overcome this we could turn to higher-level patterns of texts, such as their syntactic structure.

Syntactic stylometry was first attempted by Baayen et al. (1996), who looked at the distribution of frequencies of grammar productions.[1] More recently, Raghavan et al. (2010) identified authors by deriving a probabilistic grammar for each author and picking the author grammar that can parse the unidentified

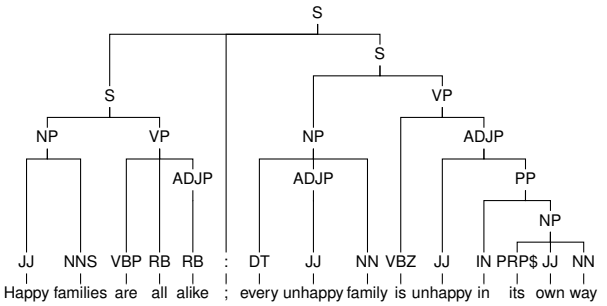

Figure 1: A phrase-structure tree produced by the Stanford parser.

text with the highest probability. There is also work that looks at syntax on a more shallow level, such as Hirst and Feiguina (2007), who work with partial parses; Wiersma et al. (2011) looked at n-grams of part-of-speech (POS) tags, and Menon and Choi (2011) focussed on particular word frequencies such as those of 'stop words,' attaining accuracies well above 90% even in cross-domain tasks.

In this work we also aim to perform syntactic stylometry, but we analyze syntactic parse trees directly, instead of summarizing the data as a set of grammar productions or a probability measure. The unit of comparison is tree fragments. Our hypothesis is that the use of fragments can provide a more interpretable model compared to one that uses fine-grained surface features such as word tokens.

2 Method

We investigate a corpus consisting of a selection of novels from a handful of authors. The corpus was selected to contain works from different time periods

[1] A grammar production is a rewrite rule that generates a constituent.

Workshop on Computational Linguistics for Literature, pages 59–63,
Montréal, Canada, June 8, 2012.

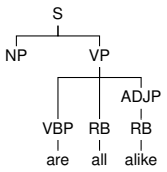

Figure 2: A phrase-structure fragment from the tree in figure 1.

from authors with a putatively distinctive style. In order to analyze the syntactic structure of the corpus we use hierarchical phrase-structures, which divide sentences into a series of constituents that are represented in a tree-structure; cf. figure 1 for an example. We analyze phrase-structures using the notion of tree fragments (referred to as subset trees by Collins and Duffy, 2002). This notion is taken from the framework of Data-Oriented Parsing (Scha, 1990), which hypothesizes that language production and comprehension exploits an inventory of fragments from previous language experience that are used as building blocks for novel sentences. In our case we can surmise that literary authors might make use of a specific inventory in writing their works, which characterizes their style. Fragments can be characterized as follows:

Definition. *A fragment f of a tree T is a connected subset of nodes from T, with $|f| \geq 2$, such that each node of f has either all or none of the children of the corresponding node in T.*

When a node of a fragment has no children, it is called a frontier node; in a parsing algorithm such nodes function as substitution sites where the fragment can be combined with other fragments. Cf. figure 2 for an example of a fragment. An important consideration is that fragments can be of arbitrary size. The notion of fragments captures anything from a single context-free production such as

(1) $S \rightarrow NP\ VP$

. . . to complete stock phrases such as

(2) Come with me if you want to live.

In other words, instead of making assumptions about grain size, we let the data decide. This is in contrast to n-gram models where n is an *a priori* defined sliding window size, which must be kept low because of data-sparsity considerations.

Author (sentences)	Works (year of first publication)
Conrad, Joseph (25,889)	Heart of Darkness (1899), Lord Jim (1900), Nostromo (1904), The Secret Agent (1907)
Hemingway, Ernest (40,818)	A Farewell To Arms (1929), For Whom the Bell Tolls (1940), The Garden of Eden (1986), The Sun Also Rises (1926)
Huxley, Aldous (23,954)	Ape and Essence (1948), Brave New World (1932), Brave New World Revisited (1958), Crome Yellow (1921), Island (1962), The Doors of Perception (1954), The Gioconda Smile (1922)
Salinger, J.D. (26,006)	Franny & Zooey (1961), Nine Stories (1953), The Catcher in the Rye (1951), Short stories (1940–1965)
Tolstoy, Leo (66,237)	Anna Karenina (1877); transl. Constance Garnett, Resurrection (1899); transl. Louise Maude, The Kreutzer Sonata and Other Stories (1889); transl. Benjamin R. Tucker, War and Peace (1869); transl. Aylmer Maude & Louise Maude

Table 1: Works in the corpus. Note that the works by Tolstoy are English translations from project Gutenberg; the translations are contemporaneous with the works of Conrad.

To obtain phrase-structures of the corpus we employ the Stanford parser (Klein and Manning, 2003), which is a treebank parser trained on the Wall Street journal (WSJ) section of the Penn treebank (Marcus et al., 1993). This unlexicalized parser attains an accuracy of 85.7 % on the WSJ benchmark ($|w| \leq 100$). Performance is probably much worse when parsing text from a different domain, such as literature; for example dialogue and questions are not well represented in the news domain on which the parser is trained. Despite these issues we expect that useful information can be extracted from the latent hierarchical structure that is revealed in parse trees, specifically in how patterns in this structure recur across different texts.

We pre-process all texts manually to strip away dedications, epigraphs, prefaces, tables of contents, and other such material. We also verified that no occurrences of the author names remained.[2] Sentence and word-level tokenization is done by the Stanford parser. Finally, the parser assigns the most likely parse tree for each sentence in the corpus. No further training is performed; as our method is memory-based, all computation is done during classification.

In the testing phase the author texts from the training sections are compared with the parse trees of texts to be identified. To do this we modified the fragment extraction algorithm of Sangati et al. (2010) to identify the common fragments among two different sets of parse trees.[3] This is a tree kernel method (Collins and Duffy, 2002) which uses dynamic programming to efficiently extract the maximal fragments that two trees have in common. We use the variant reported by Moschitti (2006) which runs in average linear time in the number of nodes in the trees.

To identify the author of an unknown text we collect the fragments which it has in common with each known author. In order to avoid biases due to different sizes of each author corpus, we use the first 15,000 sentences from each training section. From these results all fragments which were found in more than one author corpus are removed. The remaining fragments which are unique to each author are used to compute a similarity score.

We have explored different variations of similarity scores, such as the number of nodes, the average number of nodes, or the fragment frequencies. A simple method which appears to work well is to count the total number of content words.[4] Given the parse trees of a known author A and those of an unknown author B, with their unique common fragments denoted as $A \sqcap B$, the resulting similarity is defined as:

$$f(A, B) = \sum_{x \in A \sqcap B} \text{content\_words}(x)$$

However, while the number of sentences in the training sets has been fixed, they still diverge in the average number of words per sentence, which is reflected in the number of nodes per tree as well. This causes a bias because statistically, there is a higher chance that some fragment in a larger tree will match with another. Therefore we also normalize for the average number of nodes. The author can now be guessed as:

$$\arg\max_{A \in \text{Authors}} \frac{f(A, B)}{1/|A| \sum_{t \in A} |t|}$$

Note that working with content words does not mean that the model reduces to an n-gram model, because fragments can be discontiguous; e.g., "he said X but Y." Furthermore the fragments contain hierarchical structure while n-grams do not. To verify this contention, we also evaluate our model with trigrams instead of fragments. For this we use trigrams of word & part-of-speech pairs, with words stemmed using Porter's algorithm. With trigrams we simply count the number of trigrams that one text shares with another. Raghavan et al. (2010) have observed that the lexical information in n-grams and the structural information from a PCFG perform a complementary role, achieving the highest performance when both are combined. We therefore also evaluate with a combination of the two.

3 Evaluation & Discussion

Our data consist of a collection of novels from five authors. See table 1 for a specification. We perform cross-validation on 4 works per author. We evaluate on two different test sizes: 20 and 100 sentences. We test with a total of 500 sentences per work, which gives 25 and 5 datapoints per work given these sizes. As training sets only the works that are not tested on are presented to the model. The training sets consist of 15,000 sentences taken from the remaining works. Evaluating the model on these test sets took about half an hour on a machine with 16 cores, employing less than 100 MB of memory per process. The similarity functions were explored on a development set, the results reported here are from a separate test set.

The authorship attribution results are in table 2. It is interesting to note that even with three different translators, the work of Tolstoy can be successfully identified; i.e., the style of the author is modelled, not the translator's.

[2] Exception: War and Peace contains a character with the same name as its author. However, since this occurs in only one of the works, it cannot affect the results.

[3] The code used in the experiments is available at `http://github.com/andreasvc/authident`.

[4] Content words consist of nouns, verbs, adjectives, and adverbs. They are identified by the part-of-speech tags that are part of the parse trees.

20 sentences	trigrams	fragments	combined	100 sentences	trigrams	fragments	combined
Conrad	83.00	87.00	94.00	Conrad	100.00	100.00	100.00
Hemingway	77.00	52.00	81.00	Hemingway	100.00	100.00	100.00
Huxley	86.32	75.79	86.32	Huxley	89.47	78.95	89.47
Salinger	93.00	86.00	94.00	Salinger	100.00	100.00	100.00
Tolstoy	77.00	80.00	90.00	Tolstoy	95.00	100.00	100.00
average:	83.23	76.16	**89.09**	average:	96.97	95.96	**97.98**

Table 2: Accuracy in % for authorship attribution with test texts of 20 or 100 sentences.

	Conrad	Hemingway	Huxley	Salinger	Tolstoy
Conrad	**94**	1		2	3
Hemingway	3	**81**		11	5
Huxley	5	2	**82**	1	5
Salinger	1	2	3	**94**	
Tolstoy	8		2		**90**

Table 3: Confusion matrix when looking at 20 sentences with trigrams and fragments combined. The rows are the true authors, the columns the predictions of the model.

Gamon (2004) also classifies chunks of 20 sentences, but note that in his methodology data for training and testing includes sentences from the same work. Recognizing the same work is easier because of recurring topics and character names.

Grieve (2007) uses opinion columns of 500–2,000 words, which amounts to 25–100 sentences, assuming an average sentence length of 20 words. Most of the individual algorithms in Grieve (2007) score much lower than our method, when classifying among 5 possible authors like we do, while the accuracies are similar when many algorithms are combined into an ensemble. Although the corpus of Grieve is carefully controlled to contain comparable texts written for the same audience, our task is not necessarily easier, because large differences within the works of an author can make classifying that author more challenging.

Table 3 shows a confusion matrix when working with 20 sentences. It is striking that the errors are relatively asymmetric: if A is often confused with B, it does not imply that B is often confused with A. This appears to indicate that the similarity metric has a bias towards certain categories which could be removed with a more principled model.

Here are some examples of sentence-level and productive fragments that were found:

(3) Conrad: [PP [IN] [NP [NP [DT] [NN sort]] [PP [IN of] [NP [JJ] [NN]]]]]

(4) Hemingway: [VP [VB have] [NP [DT a] [NN drink]]]

(5) Salinger: [NP [DT a] [NN] [CC or] [NN something]]

(6) Salinger: [ROOT [S [NP [PRP I]] [VP [VBP mean] [SBAR]] [. .]]]

(7) Tolstoy: [ROOT [SINV [“ “] [S] [, ,] [” ”] [VP [VBD said]] [NP] [, ,] [S [VP [VBG shrugging] [NP [PRP$ his] [NNS shoulders]]]] [. .]]]

It is likely that more sophisticated statistics, for example methods used for collocation detection, or general machine learning methods to select features such as support vector machines would allow to select only the most characteristic fragments.

4 Conclusion

We have presented a method of syntactic stylometry that is conceptually simple—we do not resort to sophisticated statistical inference or an ensemble of algorithms—and takes sentence-level hierarchical phenomena into account. Contrary to much previous work in stylometry, we worked with content words rather than just function words. We have demonstrated the feasibility of analyzing literary syntax through fragments; the next step will be to use these techniques to address other literary questions.

References

Harold Baayen, H. Van Halteren, and F. Tweedie. 1996. Outside the cave of shadows: Using syntactic annotation to enhance authorship attribution. *Literary and Linguistic Computing*, pages 121–132.

Michael Collins and Nigel Duffy. 2002. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *Proceedings of ACL*.

Michael Gamon. 2004. Linguistic correlates of style: authorship classification with deep linguistic analysis features. In *Proceedings of COLING*.

Jack Grieve. 2007. Quantitative authorship attribution: An evaluation of techniques. *Literary and Linguistic Computing*, 22(3):251–270. URL http://llc.oxfordjournals.org/content/22/3/251.abstract.

Graeme Hirst and Olga Feiguina. 2007. Bigrams of syntactic labels for authorship discrimination of short texts. *Literary and Linguistic Computing*, 22(4):405–417. URL http://llc.oxfordjournals.org/content/22/4/405.abstract.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proc. of ACL*, volume 1, pages 423–430.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):313–330.

Rohith K Menon and Yejin Choi. 2011. Domain independent authorship attribution without domain adaptation. In *Proceedings of Recent Advances in Natural Language Processing*, pages 309–315.

Alessandro Moschitti. 2006. Making tree kernels practical for natural language learning. In *Proceedings of EACL*, pages 113–120. URL http://acl.ldc.upenn.edu/E/E06/E06-1015.pdf.

Sindhu Raghavan, Adriana Kovashka, and Raymond Mooney. 2010. Authorship attribution using probabilistic context-free grammars. In *Proceedings of ACL*, pages 38–42.

Federico Sangati, Willem Zuidema, and Rens Bod. 2010. Efficiently extract recurring tree fragments from large treebanks. In *Proceedings of LREC*, pages 219–226. URL http://dare.uva.nl/record/371504.

Remko Scha. 1990. Language theory and language technology; competence and performance. In Q.A.M. de Kort and G.L.J. Leerdam, editors, *Computertoepassingen in de Neerlandistiek*, pages 7–22. LVVN, Almere, the Netherlands. Original title: *Taaltheorie en taaltechnologie; competence en performance*. Translation available at http://iaaa.nl/rs/LeerdamE.html.

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556. URL http://dx.doi.org/10.1002/asi.21001.

Wybo Wiersma, John Nerbonne, and Timo Lauttamus. 2011. Automatically extracting typical syntactic differences from corpora. *Literary and Linguistic Computing*, 26(1):107–124. URL http://llc.oxfordjournals.org/content/26/1/107.abstract.