Enhancing Authorship Attribution By Utilizing Syntax Tree Profiles

Michael Tschuggnall and Günther Specht
Institute of Computer Science, University of Innsbruck
Technikerstraße 21a, 6020 Innsbruck, Austria
{michael.tschuggnall, guenther.specht}@uibk.ac.at

Abstract

The aim of modern authorship attribution approaches is to analyze known authors and to assign authorships to previously unseen and unlabeled text documents based on various features. In this paper we present a novel feature to enhance current attribution methods by analyzing the grammar of authors. To extract the feature, a syntax tree of each sentence of a document is calculated, which is then split up into length-independent patterns using pq-grams. The mostly used pq-grams are then used to compose sample profiles of authors that are compared with the profile of the unlabeled document by utilizing various distance metrics and similarity scores. An evaluation using three different and independent data sets reveals promising results and indicate that the grammar of authors is a significant feature to enhance modern authorship attribution methods.

1 Introduction

The increasing amount of documents available from sources like publicly available literary databases often raises the question of verifying disputed authorships or assigning authors to unlabeled text fragments. The original problem was initiated already in the midst of the twentieth century by Mosteller and Wallace, who tried to find the correct authorships of *The Federalist Papers* (Mosteller and Wallace, 1964), nonetheless authorship attribution is still a major research topic. Especially with latest events in politics and academia, the verification of authorships becomes increasingly important and is used frequently in areas like juridical applications (*Forensic Linguistics*) or cybercrime detection (Nirkhi and Dharaskar, 2013). Similarily to works in the field of plagiarism detection (e.g. (Stamatatos, 2009; Tschuggnall and Specht, 2013b)) which aim to find text fragments not written but claimed to be written by an author, the problem of traditional authorship attribution is defined as follows: Given several authors with text samples for each of them, the question is to label an unknown document with the correct author. In contrast to this so-called *closed-class* problem, an even harder task is addressed in the *open-class* problem, where additionally a "none-of-them"-answer is allowed (Juola, 2006).

In this paper we present a novel feature for the traditional, closed-class authorship attribution task, following the assumption that different authors have different writing styles in terms of the grammar structure that is used mostly unconsciously. Due to the fact that an author has many different choices of how to formulate a sentence using the existing grammar rules of a natural language, the assumption is that the way of constructing sentences is significantly different for individual authors. For example, the famous Shakespeare quote *"To be, or not to be: that is the question."* (S1) could also be formulated as *"The question is whether to be or not to be."* (S2) or even *"The question is whether to be or not."* (S3) which is semantically equivalent but differs significantly according to the syntax (see Figure 1). The main idea of this approach is to quantify those differences by calculating grammar profiles for each candidate author as well as for the unlabeled document, and to assign one of the candidates as the author of the unseen document by comparing the profiles. To quantify the differences between profiles multiple metrics have been implemented and evaluated.

The rest of this paper is organized as follows: Section 2 sketches the main idea of the algorithm which incorporates the distance metrics explained in detail in Section 3. An extensive evaluation us-

Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 195–199,
Gothenburg, Sweden, April 26-30 2014.

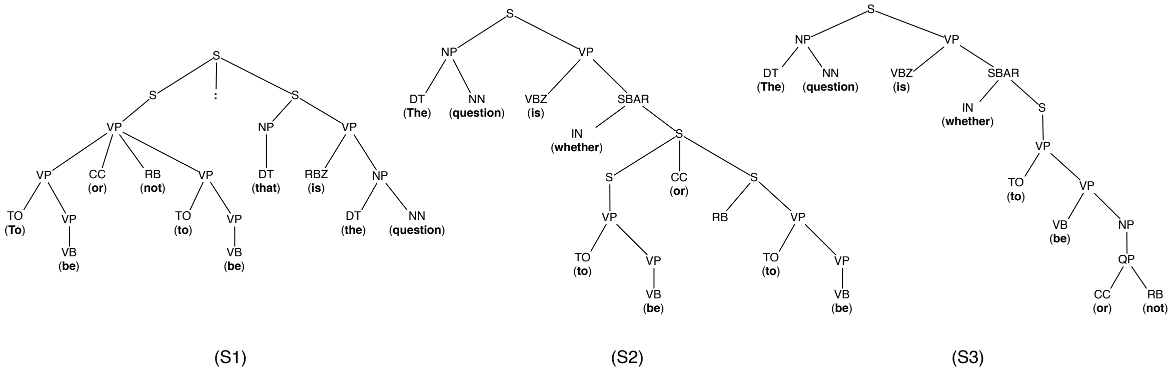

Figure 1: Syntax Trees Resulting From Parsing Sentence (S1), (S2) and (S3).

ing three different test sets is shown in Section 4, while finally Section 5 and Section 6 summarize related work and discuss future work, respectively.

2 Syntax Tree Profiles

The basic idea of the approach is to utilize the syntax that is used by authors to distinguish authorships of text documents. Based on our previous work in the field of intrinsic plagiarism detection (Tschuggnall and Specht, 2013c; Tschuggnall and Specht, 2013a) we modify and enhance the algorithms and apply them to be used in closed-class authorship attribution.

The number of choices an author has to formulate a sentence in terms of grammar is rather high, and the assumption in this approach is that the concrete choice is made mostly intuitively and unconsciously. Evaluations shown in Section 4 reinforce that solely parse tree structures represent a significant feature that can be used to distinguish between authors.

From a global view the approach comprises the following three steps: *(A)* Creating a grammar profile for each author, *(B)* creating a grammar profile for the unlabeled document, and *(C)* calculating the distance between each author profile and the document profile and assigning the author having the lowest distance (or the highest similarity, depending on the distance metric chosen). As this approach is based on profiles a key criterion is the creation of distinguishable author profiles. In order to calculate a grammar profile for an author or a document, the following procedure is applied: *(1)* Concatenate all text samples for the author into a single, large sample document, *(2)* split the resulting document into single sentences and calculate a syntax tree for each sentence, *(3)* calculate the pq-gram index for each tree, and *(4)* compose the final grammar profile from the normalized frequencies of pq-grams.

At first the concatenated document is cleaned to contain alphanumeric characters and punctuation marks only, and then split into single sentences[1]. Each sentence is then parsed[2]. For example, Figure 1 depicts the syntax trees resulting from sentences (S1), (S2) and (S3). The labels of each tree correspond to a Penn Treebank tag (Marcus et al., 1993), where e.g *NP* corresponds to a noun phrase or *JJS* to a superlative adjective. In order to examine solely the structure of sentences, the terminal nodes (words) are ignored.

Having computed a syntax tree for every sentence, the pq-gram index (Augsten et al., 2010) of each tree is calculated in the next step. Pq-grams consist of a stem (p) and a base (q) and may be related to as "n-grams for trees". Thereby p defines how much nodes are included vertically, and q defines the number of nodes to be considered horizontally. For example, a pq-gram using $p = 2$ and $q = 3$ starting from level two of tree (S1) would be [S-VP-VP-CC-RB]. In order to obtain all pq-grams of a tree, the base is additionally shifted left and right: If then less than p nodes exist horizontally, the corresponding place in the pq-gram is filled with $\star$, indicating a missing node. Applying this idea to the previous example, also the pq-grams [S-VP-⋆-⋆-VP] (base shifted left by two), [S-VP-⋆-VP-CC] (base shifted left by one), [S-VP-RB-VP-⋆] (base shifted right by one) and [S-VP-VP-⋆-⋆] (base shifted right by two) have to be considered. Finally, the pq-gram index contains all pq-grams of

[1] using OpenNLP, http://incubator.apache.org/opennlp, visited October 2013

[2] using the Stanford Parser (Klein and Manning, 2003)

a syntax tree, whereby multiple occurences of the same pq-grams are also present multiple times in the index.

The remaining part for creating the author profile is to compute the pq-gram index of the whole document by combining all pq-gram indexes of all sentences. In this step the number of occurences is counted for each pq-gram and then normalized by the total number of all appearing pq-grams. As an example, the three mostly used pq-grams of a selected document together with their normalized frequencies are {`[NP-NN-⋆-⋆-⋆]`, 2.7%}, {`[PP-IN-⋆-⋆-⋆]`, 2.3%}, and {`[S-VP-⋆-⋆-VBD]`, 1.1%}. The final *pq-gram profile* then consists of the complete table of pq-grams and their occurences in the given document.

3 Distance and Similarity Metrics

With the use of the syntax tree profiles calculated for each candidate author as well as for the unlabeled document, the last part is to calculate a distance or similarity, respectively, for every author profile. Finally, the unseen document is simply labeled with the author of the best matching profile.

To investigate on the best distance or similarity metric to be used for this approach, several metrics for this problem have been adapted and evaluated[3]: 1. CNG (Kešelj et al., 2003), 2. Stamatatos-CNG (Stamatatos, 2009), 3. Stamatatos-CNG with Corpus Norm (Stamatatos, 2007), 4. Sentence-SPI.

For the latter, we modified the original SPI score (Frantzeskou et al., 2006) so that each sentence is traversed separately: Let S_D be the set of sentences of the document, $I(s)$ the pq-gram-index of sentence s and P_x the profile of author X, then the Sentence-SPI score is calculated as follows:

$$s_{P_x,P_D} = \sum_{s \in S_D} \sum_{p \in I(s)} \begin{cases} 1 & \text{if } p \in P_x \\ 0 & \text{else} \end{cases}$$

4 Evaluation

The approach described in this paper has been extensively evaluated using three different English data sets, whereby all sets are completely unrelated and of different types: (1.) CC04: the training set used for the Ad-hoc-Authorship Attribution Competition workshop held in 2004[4] - type: novels, authors: 4, documents: 8, samples per author: 1; (2.) FED: the (undisputed) federalist papers written by Hamilton, Madison and Jay in the 18th century - type: political essays, authors: 3, documents: 61, samples per author: 3; (3.) PAN12: from the state-of-the-art corpus, especially created for the use in authorship identification for the PAN 2012 workshop[5] (Juola, 2012), all closed-classed problems have been chosen - type: misc, authors: 3-16, documents: 6-16, samples per author: 2.

For the evaluation, each of the sets has been used to optimize parameters while the remaining sets have been used for testing. Besides examining the discussed metrics and values for p and q (e.g. by choosing $p = 1$ and $q = 0$ the pq-grams of a grammar profile are equal to pure POS tags), two additional optimization variables have been integrated for the similarity metric Sentence-SPI:

- **topPQGramCount** t_c**:** by assigning a value to this parameter, only the corresponding amount of mostly used pq-grams of a grammar profile are used.
- **topPQGramOffset** t_o**:** based on the idea that all authors might have a frequently used and common set of syntax rules that are predefined by a specific language, this parameter allows to ignore the given amount of mostly used pq-grams. For example if $t_o = 3$ in Table 1, the first pq-gram to be used would be `[NP-NNP-⋆-⋆-⋆]`.

The evaluation results are depicted in Table 1. It shows the rate of correct author attributions based on the grammar feature presented in this paper.

Generally, the algorithm worked best using the *Sentence-SPI* score, which led to a rate of 72% by using the PAN12 data set for optimization. The optimal configuration uses $p = 3$ and $q = 2$, which is the same configuration that was used in (Augsten et al., 2010) to produce the best results. The highest scores are gained by using a limit of top pq-grams ($t_c \sim 65$) and by ignoring the first three pq-grams ($t_o = 3$), which indicates that it is sufficient to limit the number of syntax structures

[3] The algorithm names are only used as a reference for this paper, but were not originally proposed like that

[4] http://www.mathcs.duq.edu/∼juola/authorship_contest.html, visited Oct. 2013

[5] PAN is a well-known workshop on Uncovering Plagiarism, Authorship, and Social Software Misuses. http://pan.webis.de, visited Oct. 2013

metric	p	q	Optimized With	CC04	FED	PAN12	Overall
Sentence-SPI ($t_c = 65, t_o = 3$)	3	2	PAN12	57.14	86.89	*(76.04)*	**72.02**
CNG	0	2	PAN12	14.29	80.33	*(57.29)*	**47.31**
Stamatatos-CNG	2	2	PAN12	14.29	78.69	*(60.42)*	**46.49**
Stamatatos-CNG-CN	0	2	CC04	*(42.86)*	52.46	18.75	**35.61**

Table 1: Evaluation Results.

and that there exists a certain number (3) of general grammar rules for English which are used by *all* authors. I.e. those rules cannot by used to infer information about individual authors (e.g. every sentence starts with `[S-...]`).

All other metrics led to worse results, which may also be a result of the fact that only the Sentence-SPI metric makes use of the additional parameters t_c and t_o. Future work should also investigate on integrating these parameters also in other metrics. Moreover, results are better using the PAN12 data set for optimization, which may be because this set is the most hetergeneous one: The Federalist Papers contain only political essays written some time ago, and the CC04 set only uses literary texts written by four authors.

5 Related Work

Successful current approaches often are based on or include character n-grams (e.g. (Hirst and Feiguina, 2007; Stamatatos, 2009)). Several studies have shown that n-grams represent a significant feature to identify authors, whereby the major benefits are the language independency as well as the easy computation. As a variation, word n-grams are used in (Balaguer, 2009) to detect plagiarism in text documents.

Using individual features, machine learning algorithms are often applied to learn from author profiles and to predict unlabeled documents. Among methods that are utilized in authorship attribution as well as the related problem classes like text categorization or intrinsic plagiarism detection are support vector machines (e.g. (Sanderson and Guenter, 2006; Diederich et al., 2000)), neural networks (e.g. (Tweedie et al., 1996)), naive bayes classifiers (e.g. (McCallum and Nigam, 1998)) or decision trees (e.g. (Ö. Uzuner et. al, 2005)).

Another interesting approach used in authorship attribution that tries to detect the writing style of authors by analyzing the occurences and variations of spelling errors is proposed in (Koppel and Schler, 2003). It is based on the assumption that authors tend to make similar spelling and/or grammar errors and therefore uses this information to attribute authors to unseen text documents.

Approaches in the field of genre categorization also use NLP tools to analyze documents based on syntactic annotations (Stamatatos et al., 2000). Lexicalized tree-adjoining-grammars (LTAG) are poposed in (Joshi and Schabes, 1997) as a ruleset to construct and analyze grammar syntax by using partial subtrees.

6 Conclusion and Future Work

In this paper we propose a new feature to enhance modern authorship attribution algorithms by utilizing the grammar syntax of authors. To distinguish between authors, syntax trees of sentences are calculated which are split into parts by using pq-grams. The set of pq-grams is then stored in an author profile that is used to assign unseen documents to known authors.

The algorithm has been optimized and evaluated using three different data sets, resulting in an overall attribution rate of 72%. As the work in this paper solely used the grammar feature and completely ignores information like the vocabulary richness or n-grams, the evaluation results are promising. Future work should therefore concentrate on integrating other well-known and good-working features as well as considering common machine-learning techniques like support vector machines or decision trees to predict authors based on pq-gram features. Furthermore, the optimization parameters currently only applied on the similiarity score should also be integrated with the distance metrics as they led to the best results. Research should finally also be done on the applicability to other languages, especially as syntactically more complex languages like German or French may lead to better results due to the higher amount of grammar rules, making the writing style of authors more unique.

References

Nikolaus Augsten, Michael Böhlen, and Johann Gamper. 2010. The pq-Gram Distance between Ordered Labeled Trees. *ACM Transactions on Database Systems (TODS).*

Enrique Vallés Balaguer. 2009. Putting Ourselves in SME's Shoes: Automatic Detection of Plagiarism by the WCopyFind tool. In *Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse*, pages 34–35.

Joachim Diederich, Jörg Kindermann, Edda Leopold, and Gerhard Paass. 2000. Authorship attribution with support vector machines. *APPLIED INTELLIGENCE*, 19:2003.

Georgia Frantzeskou, Efstathios Stamatatos, Stefanos Gritzalis, and Sokratis Katsikas. 2006. Effective identification of source code authors using byte-level information. In *Proceedings of the 28th international conference on Software engineering*, pages 893–896. ACM.

Graeme Hirst and Ol'ga Feiguina. 2007. Bigrams of syntactic labels for authorship discrimination of short texts. *Literary and Linguistic Computing*, 22(4):405–417.

Aravind K. Joshi and Yves Schabes. 1997. Tree-adjoining grammars. In *Handbook of formal languages*, pages 69–123. Springer.

Patrick Juola. 2006. Authorship attribution. *Foundations and Trends in Information Retrieval*, 1(3):233–334.

Patrick Juola. 2012. An overview of the traditional authorship attribution subtask. In *CLEF (Online Working Notes/Labs/Workshop).*

Vlado Kešelj, Fuchun Peng, Nick Cercone, and Calvin Thomas. 2003. N-gram-based author profiles for authorship attribution. In *Proceedings of the conference pacific association for computational linguistics, PACLING*, volume 3, pages 255–264.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 423–430, Stroudsburg, PA, USA.

Moshe Koppel and Jonathan Schler. 2003. Exploiting Stylistic Idiosyncrasies for Authorship Attribution. In *IJCAI'03 Workshop On Computational Approaches To Style Analysis And Synthesis*, pages 69–72.

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

Andrew McCallum and Kamal Nigam. 1998. A comparison of event models for naive bayes text classification.

F. Mosteller and D. Wallace. 1964. *Inference and Disputed Authorship: The Federalist.* Addison-Wesley.

Smita Nirkhi and RV Dharaskar. 2013. Comparative study of authorship identification techniques for cyber forensics analysis. *International Journal.*

Ö. Uzuner et. al. 2005. Using Syntactic Information to Identify Plagiarism. In *Proc. 2nd Workshop on Building Educational Applications using NLP.*

Conrad Sanderson and Simon Guenter. 2006. Short text authorship attribution via sequence kernels, markov chains and author unmasking: an investigation. In *Proc. of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 482–491, Stroudsburg, PA, USA.

Efstathios Stamatatos, George Kokkinakis, and Nikos Fakotakis. 2000. Automatic text categorization in terms of genre and author. *Comput. Linguist.*, 26:471–495, December.

Efstathios Stamatatos. 2007. Author identification using imbalanced and limited training texts. In *Database and Expert Systems Applications, 2007. DEXA'07. 18th International Workshop on*, pages 237–241. IEEE.

Efstathios Stamatatos. 2009. Intrinsic Plagiarism Detection Using Character n-gram Profiles. In *CLEF (Notebook Papers/Labs/Workshop).*

Michael Tschuggnall and Günther Specht. 2013a. Countering Plagiarism by Exposing Irregularities in Authors Grammars. In *EISIC, European Intelligence and Security Informatics Conference, Uppsala, Sweden*, pages 15–22.

Michael Tschuggnall and Günther Specht. 2013b. Detecting Plagiarism in Text Documents through Grammar-Analysis of Authors. In *15. GI-Fachtagung Datenbanksysteme für Business, Technologie und Web, Magdeburg, Germany.*

Michael Tschuggnall and Günther Specht. 2013c. Using grammar-profiles to intrinsically expose plagiarism in text documents. In *NLDB*, pages 297–302.

Fiona J. Tweedie, S. Singh, and David I. Holmes. 1996. Neural network applications in stylometry: The federalist papers. *Computers and the Humanities*, 30(1):1–10.