Authorship Verification based on Syntax Features

Jan Rygl, Kristýna Zemková, Vojtěch Kovář

NLP Centre
Faculty of Informatics
Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
`xrygl@fi.muni.cz`

Abstract. Authorship verification is wildly discussed topic at these days. In the authorship verification problem, we are given examples of the writing of an author and are asked to determine if given texts were or were not written by this author. In this paper we present an algorithm using syntactic analysis system SET for verifying authorship of the documents. We propose three variants of two-class machine learning approach to authorship verification. Syntactic features are used as attributes in suggested algorithms and their performance is compared to established word-lenth distribution features. Results indicate that syntactic features provide enough information to improve accuracy of authorship verification algorithms.

Key words: authorship verification, syntactic analysis

1 Introduction

The interest in autorship verification can be found in 18th century in the Shakespearean era. A lot of linguists wanted to prove (or disprove) that William Shakespeare wrote the well known plays [1]. After that, this topic was discussed more and more often.

The task of authorship verification is commonly distinguished from that of authorship attribution. In both text classification approaches, the task is to decide whether a given text has been written by a candidate author. In authorship attribution, the actual author is known to be included in the set of candidates (closed case). In authorship verification, however, this assumption cannot be made: the given text might have been written by one of the candidate authors, but could also be written by none of them (open case). Note that this scenario is typical of forensic applications where it cannot be presupposed that the author of, for example, a letter bomb is among the suspect candidate authors.[2]

There are three main approaches to the authorship verification:

1. One-Class Machine Learning: In this approach[3] authors used only positive examples for training, because they consider difficult to select representative negative examples.

Aleš Horák, Pavel Rychlý (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2012, pp. 111–119, 2012. © Tribun EU 2012

2. Two-Class Machine Learning: The technique is established for an authorship attribution task.
3. Unmasking algorithm: The main assumption is that only small number of features distinguish between authors. The most distinguishing features are iteratively removed. Hypothesis is that whatever differences there are between document will be reflected in only a relatively small number of features. [4]

So far, not many authors have specialized their work to authorship verification, especially with syntactic features. The availability of fast and accurate natural language parsers allow for serious research into syntactic stylometry. [5]

In this paper, we focus on syntactic features combined with the Two-Class Machine Learning approach. The unmasking algorithm requires several different types of features and One-Class Machine Learning performs worse than Two-Class ML [3]. Three implementations of Two-Class ML approach are tested. The basic variant using Support Vector Machines is utilized and two modifications are suggested:

- The authorship verification problem is converted to the authorship attribution task by adding several random documents.
- In the authorship attribution problem, similarities of documents are transformed to rankings of documents. [6]

Because, unlike other publications, we work with texts consisting of up to tens of sentences, we have to cope with insufficiency of qualitative and quantitative information. Not many linguistics have focused on short texts because not enough material can cause lower accuracy.

2 Syntactic Analysis of the Czech Language

The main aim of the natural language syntactic analysis is to show the surface structure of the input sentence. Czech is one of the free-word-order languages with rich morphology that poses barriers to parsing using formalisms that are relatively succesfull when used with fixed-word-order languages such as English. Because of unrestricted word order in Czech, current Czech parsers face problems such as high ambiguity of the analysis output or low precision or coverage on corpus texts.

There are three main approaches to the automatic syntactic analysis of Czech at this time. The first uses the formalism of Functional Generative Description, FGD, for syntax description and is developed at the Institute of Formal and Applied Linguistics in Prague. Within this formalism, the syntactic information is encoded as an acyclic connected graph of dependency relations, called *dependency tree*. [7]

The second approach to Czech language parsing, `synt`, uses the constituent formalism of syntax and its development centre is located at the Natural Language Processing Centre of Masaryk University in Brno. The constituent formalism encodes the syntactic information as a derivation tree based on the formal grammar of the language. System `synt` is based on a metagrammar formalism with a context-free backbone, contextual actions and an efficient variant of the chart parsing algorithm. [8]

The third approach is system SET. This open source tool was developed at the NLP Centre as well. SET is based on the simple principle of pattern matching, so it is fast, understandable for people and easily extensible. It is written in Python which means it is easily usable on different platforms and there is no need for complicated installation. The core of SET consists of a set of patterns (or rules) and a pattern matching engine that analyses the input sentence according to given rules. Currently, SET is distributed with a set of rules for parsing the Czech language, containing about 100 rules. The primary output of the analysis is a *hybrid tree* – a combination of constituent and dependency formalism – but SET also offers converting this tree into purely dependency or purely constituent formalism. Other output options include extraction of phrases in several settings, finding dependencies among these phrases or extraction of collocations.

3 Extracting Syntax Features using SET

Nowadays, SET is one of the fastest available parsing systems for Czech with reasonable precision, it is freely available and very easy to use. Therefore we decided to use it for extraction of syntactic features in our experiment. As outlined above, SET produces parsing trees in three possible output fomats: dependency format (`-d` option), constituent format (`-p` option) and hybrid format (default). Dependency and constituent tree is illustrated in Figure 1, for Czech sentence *Verifikujeme autorství se syntaktickou analýzou.* (*We verify the authorship using syntactic analysis.*), as analyzed by SET. On the left hand side, we can see a phrasal tree; on the right side, a dependency tree.

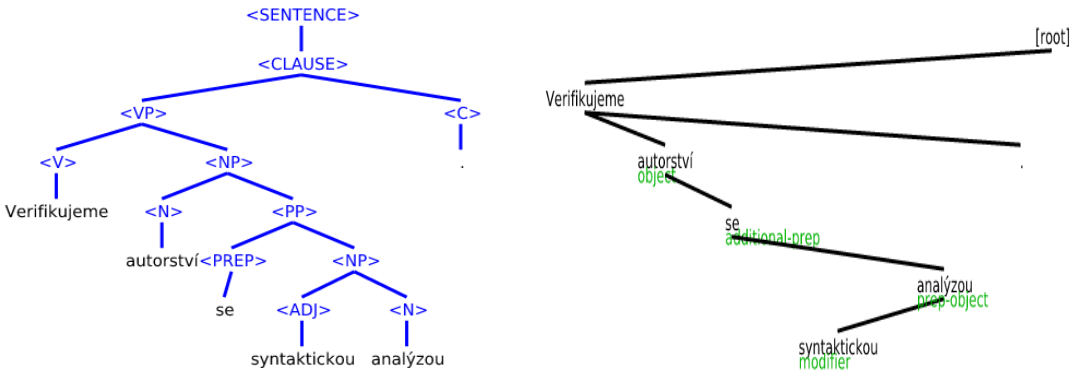

Fig. 1. Dependency and phrasal tree from SET

The dependency and phrasal output of SET was used to extract features for machine learning of differences among the authors. Namely, the following features were used:

- maximum depth of the dependency tree
- highest number of child nodes in the dependency tree
- absolute and relative frequencies of particular non-terminals in the phrasal tree (e.g. *<CLAUSE>*, *<NP>*, *<VP>*)
- absolute and relative frequencies of particular dependency labels in the dependency tree (e.g. *prep-object*, *verb-object*)

4 Authorship Verification Algorithms

In authorship verification problem, we are given two documents A and B and are asked to determine if documents were or were not written by the same author.

Two-Class Machine Learning algorithm was implemented and other two algorithms were designed to verify that two documents were written by the same author.

1. **Two-Class Machine Learning:**
   Basic approach to Authorship Verification is to train Machine Learning model to decide if two documents A, B do or do not have the same author. The main disadvantage is that it is impossible to cover all types of negative examples in training data.

```
given $document_A$, $document_B$, empty $attributeList$
for $i$ in $1 \ldots count(features)$:
        $feature = features[i]$
        $attributeList[i] = |feature(document_A) - feature(document_B)|$
$Model(attributeList)$ predicts if documents were written by same author.
```

2. **Converting verification to attribution problem:**
   "Authorship verification . . . generally deemed more difficult than so-called authorship attribution."[2], therefore we transformed problem by adding 4 documents D_1, . . . , D_4. Attribution method selects from candidates B, D_1, . . . , D_4 the most similar document to A. If the document B is selected with enough confidence, documents A and B are written by same author.

```
given $document_A$, $document_B$, empty $attributeList$
select 4 random documents $(D_1, D_2, D_3, D_4)$ of similar length to $document_B$
for $doc$ in $(document_B, D_1, D_2, D_3, D_4)$:
        empty $attributeList$
        for $i$ in $1 \ldots count(features)$:
              $feature = features[i]$
              $attributeList[i] = |feature(document_A) - feature(doc)|$
        $Model(attributeList)$ computes probability $prob_{doc}$ of same authorship
```

```
if $prob_B >= 0.5 \wedge prob_B$ = max($prob_{B,1,2,3,4}$): "same authorship"
```

3. **Algorithm 2 extended by replacing similarity scores by their rankings:** Our previous experiments showed that accuracy of Authorship Attribution problem can be improved by replacing similarities of documents by their rankings. [6]

```
given $document_A$, $document_B$, empty attributeList
select 4 random documents ($D_1, D_2, D_3, D_4$) of similar length to $document_B$
for i in 1 ...count(features):
        feature = features[i]
        rank $= 1$
        diff$= |$feature($document_A$)$-$ feature($document_B$)$|$
        for doc in ($D_1, D_2, D_3, D_4$):
              if $|$feature($document_A$)$-$ feature(doc)$|$ $<$diff: rank $+ =$
1
        attributeList[i] $= \frac{1}{rank}$
Model(attributeList) predicts if documents were written by same author.
```

5 Experiments

Data

400 Czech documents (10 documents per author) downloaded from the Internet were used. The data were collected from Czech blogs and Internet discussions connected to these blogs and were preprocessed automatically by the Czech morphological tagger *Desamb* [9] and the *SET* parser [7]. The document length ranges from 1 to about 100 sentences.

Machine Learning

LIBSVM [10] implementation of Support Vector Machines algorithm was selected as the machine learning component and 4-fold cross-validation was used for evaluation.

Authors were divided into 4 groups, each group contained 10 authors and 100 documents. During all experiments, authors of learning documents were different to authors of test documents.

Models were trained utilizing 1000 positive and 1000 negative examples for each scenario. To create positive examples, documents A and B were randomly selected from the same author; to simulate negative examples, an author of document B was different to the author of A. Authors of documents $D_1, \ldots, D_4$ used in algorithm 2 and 3 were different to authors of A and B for both positive and negative examples.

Algorithm 1: Two-Class ML

For the basic algorithm, the average accuracy was **57.9 %** (7.9 % over the baseline). Detailed results are shown in Table 1.

Table 1. Results of Algorithm 1

(a) Folder 1: Accuracy: 51.1 %

	Positive	Negative
True	280 (38.5 %)	92 (12.6 %)
False	272 (37.4 %)	84 (11.5 %)

(b) Folder 2: Accuracy: 55.4 %

	Positive	Negative
True	360 (41.7 %)	119 (13.8 %)
False	313 (36.2 %)	72 (8.3 %)

(c) Folder 3: Accuracy: 67.7 %

	Positive	Negative
True	230 (33.6 %)	233 (34.1 %)
False	109 (15.9 %)	112 (16.4 %)

(d) Folder 4: Accuracy: 57.2 %

	Positive	Negative
True	224 (28.7 %)	222 (28.5 %)
False	168 (21.5 %)	166 (21.3 %)

```
Folder 1: Train accuracy 77.4 % for parameters c=2.0 g=0.5
Folder 2: Train accuracy 75.5 % for parameters c=8.0 g=0.5
Folder 3: Train accuracy 70.2 % for parameters c=2048.0 g=0.125
Folder 4: Train accuracy 73.3 % for parameters c=2048.0 g=0.125
```

Algorithm 2: Converting Authorship Verification to Attribution

This method was found to be unsuitable to solve Authorship Verification problem. Average accuracy did not even exceed the baseline.

Algorithm 3: Converting Authorship Verification to Attribution using Ranking instead of Score

With the last algorithm, the average accuracy was **71.3 %**. If we consider short lengths of documents, obtained results are good. Accuracy of this method is 21.3 % better than the baseline and represents 13.5 % improvement over algorithm 1). See detailed results in Table 2.

Performance Comparison: Word-Length Distribution

Word-Length approach published by T. C. Mendenhall in 1887 [11] is still used in many current works. To compare our syntactic features with this approach, we replaced them by the word-length distribution and then used the same algorithms.

- **Algorithm 1:** Two-Class ML with Word-Length Features
  Average accuracy is **53.2 %**, which is only slightly better than the baseline. Detailed results are shown in Table 3.

Table 2. Results of Algorithm 3

(a) Folder 1: Accuracy: 79.3 %

	Positive	Negative
True	691 (34.6 %)	894 (44.7 %)
False	106 (5.3 %)	309 (15.4 %)

(b) Folder 2: Accuracy: 64.3 %

	Positive	Negative
True	364 (18.2 %)	921 (46.0 %)
False	79 (4.0 %)	636 (31.8 %)

(c) Folder 3: Accuracy: 69.0 %

	Positive	Negative
True	481 (24.1 %)	899 (44.9 %)
False	101 (5.1 %)	519 (25.9 %)

(d) Folder 4: Accuracy: 72.8 %

	Positive	Negative
True	491 (24.6 %)	965 (48.2 %)
False	35 (1.8 %)	509 (25.4 %)

```
Folder 1: Train accuracy 88.9 % for parameters c=512.0 g=0.125
Folder 2: Train accuracy 88.2 % for parameters c=2048.0 g=2.0
Folder 3: Train accuracy 88.0 % for parameters c=8.0 g=2.0
Folder 4: Train accuracy 87.7 % for parameters c=8.0 g=2.0
```

Table 3. Results of Algorithm 1

(a) Folder 1: Accuracy: 52.9 %

	Positive	Negative
True	404 (44.9 %)	72 (8.0 %)
False	378 (42.0 %)	46 (5.1 %)

(b) Folder 2: Accuracy: 53.0 %

	Positive	Negative
True	358 (39.8 %)	119 (13.2 %)
False	331 (36.8 %)	92 (10.2 %)

(c) Folder 3: Accuracy: 50.1 %

	Positive	Negative
True	326 (36.2 %)	125 (13.9 %)
False	325 (36.1 %)	124 (13.8 %)

(d) Folder 4: Accuracy: 56.9 %

	Positive	Negative
True	358 (39.8 %)	154 (17.1 %)
False	296 (32.9 %)	92 (10.2 %)

```
Folder 1: Train accuracy 77.8 % for parameters c=8.0 g=0.125
Folder 2: Train accuracy 77.9 % for parameters c=2.0 g=0.5
Folder 3: Train accuracy 80.0 % for parameters c=2.0 g=0.5
Folder 4: Train accuracy 79.4 % for parameters c=8192.0 g=0.0078125
```

– **Algorithm 3:** Authorship Attribution with Rankings replacing Scores (with Word-Length Features)

  Average accuracy is **61.5 %**. The train accuracies indicate that the machine learning model is partially overfitted. The accuracy could be slightly increased by further optimizations, involving heuristic selection of attributes, but given described size of the learning set and lengths of documents, word-length features are outperformed by our syntactic features. Results are displayed in Table 4.

Table 4. Results of Algorithm 3

(a) Folder 1: Accuracy: 62.7 %

	Positive	Negative
True	259 (12.9 %)	994 (49.7 %)
False	6 (0.3 %)	741 (37.1 %)

(b) Folder 2: Accuracy: 61.4 %

	Positive	Negative
True	229 (11.4 %)	998 (49.9 %)
False	2 (0.1 %)	771 (38.6 %)

(c) Folder 3: Accuracy: 62.2 %

	Positive	Negative
True	244 (12.2 %	999 (49.9 %)
False	1 (0.1 %)	756 (37.8 %)

(d) Folder 4: Accuracy: 59.8 %

	Positive	Negative
True	196 (9.8 %)	1000 (50.0 %)
False	0 (0.0 %)	804 (40.2 %)

```
Folder 1: Train accuracy 91.9 % for parameters c=2.0 g=2.0
Folder 2: Train accuracy 91.3 % for parameters c=2.0 g=2.0
Folder 3: Train accuracy 91.1 % for parameters c=8.0 g=2.0
Folder 4: Train accuracy 90.7 % for parameters c=8.0 g=2.0
```

6 Conclusions and Future Work

The primary aim of this paper was to present a syntactic approach to the authorship verification task. Because, unlike other publications, we work with texts consisting of up to tens of sentences, we have to cope with insufficiency of qualitative and quantitative information. Despite the fact that the accuracy of our method does not achieve desired results yet, the experiment indicates that syntactic features can outperform established approaches.

Within the future work, our goal is to find another syntactic atributes to add to our algorithms. We also plan combining syntactical and morphological information together.

Acknowledgments

This work has been partly supported by the Ministry of the Interior of CR within the project VF20102014003.

References

1. Malone, Edmond: A Dissertation on the Three Parts of King Henry VI. Tending to Shew That Those Plays Were Not Written Originally by Shakspeare. Gale Ecco, Print Editions (1787)
2. Kestemont, M., Luyckx, K., Daelemans, W., Crombez, T.: Cross-genre authorship verification using unmasking. English Studies **93**(3) (2012) 340–356
3. Manevitz, L.M., Yousef, M., Cristianini, N., Shawe-taylor, J., Williamson, B.: One-class svms for document classification. Journal of Machine Learning Research **2** (2001) 139–154

4. Koppel, M., Schler, J.: Authorship verification as a one-class classification problem. In: Proceedings of the twenty-first international conference on Machine learning. ICML '04, New York, NY, USA, ACM (2004) 62–
5. Hollingsworth, C.: Using dependency-based annotations for authorship identification. In Sojka, P., Horák, A., Kopeček, I., Pala, K., eds.: Text, Speech and Dialogue. Volume 7499 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2012) 314–319
6. Rygl, J., Horák, A.: Similarity ranking as attribute for machine learning approach to authorship identification. In Chair), N.C.C., Choukri, K., Declerck, T., Doğan, M.U., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., eds.: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey, European Language Resources Association (ELRA) (2012)
7. Kovář, V., Horák, A., Jakubíček, M.: Syntactic analysis using finite patterns: A new parsing system for czech. In Vetulani, Z., ed.: LTC. Volume 6562 of Lecture Notes in Computer Science., Springer (2009) 161–171
8. Horák, A.: Computer Processing of Czech Syntax and Semantics. Librix.eu, Brno, Czech Republic (2008)
9. Šmerk, Pavel: K počítačové morfologické analýze češtiny. PhD thesis, Faculty of Informatics Masaryk University (2010)
10. Chang, Chih-Chung - Lin, Chih-Jen: LIBSVM: a library for support vector machines. (2001) URL: `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.
11. Mendenhall, T. C.: The characteristic curves of composition. The Popular Science **11** (1887) 237–246