A Comparative Study of Language Modeling to Instance-Based Methods, and Feature Combinations for Authorship Attribution

Olga Fourkioti, Symeon Symeonidis(✉), and Avi Arampatzis

Database and Information Retrieval Research Unit,
Department of Electrical and Computer Engineering,
Democritus University of Thrace, 67100 Xanthi, Greece
olgafour1@gmail.com, {ssymeoni,avi}@ee.duth.gr
http://www.nonrelevant.net

Abstract. We present a comparative study of language modeling to traditional instance-based methods for authorship attribution, using several different basic units as features, such as characters, words, and other simple lexical measurements, as well as we propose the use of part-of-speech (POS) tags as features for language modeling. In contrast to many other studies which focus on small sets of documents written by major writers regarding several topics, we consider a relatively large corpus with documents edited by non-professional writers regarding the same topic. We find that language models based on either characters or POS tags are the most effective, while the latter provide additional efficiency benefits and robustness against data sparsity. Moreover, we experiment with linearly combining several language models, as well as employing unions of several different feature types in instance-based methods. We find that both such combinations constitute viable strategies which generally improve effectiveness. By linearly combining three language models, based respectively on character, word, and POS trigrams, we achieve the best generalization accuracy of 96%.

Keywords: Authorship attribution · Text mining · Language models · Computational linguistics · Text categorization · Text classification · Machine learning

1 Introduction

Authorship attribution can be used in a broad range of applications. Apart from literary research where a document of disputed authorship is assigned to one candidate author, the plethora of anonymous electronic texts (e.g. emails, blogs, electronic messages, forums, source code, etc.) has rendered authorship attribution analysis indispensable to diverse areas dealing with real-world texts. These areas include civil law (e.g. in cases of disputed copyrights), criminal law (e.g. in order to identify the author of a suicidal note or a terrorist's proclamation),

J. Kamps et al. (Eds.): TPDL 2017, LNCS 10450, pp. 274–286, 2017.
DOI: 10.1007/978-3-319-67008-9_22

forensics (e.g. in order to determine the author of source code of malignant software), and others.

The general approach to authorship attribution is based on the extraction of features that have a high discriminatory potential between the candidate authors, the so-called style markers, which is followed by a feature selection method and the training of a classifier [13]. Specifically, there are three main types of style markers reflecting a document's representation: lexical, character, and syntactic [18]. Current authorship attribution studies are dominated by lexical and character feature types meaning that a document is considered as a sequence of words or characters respectively.

There have been proposed many measures in an attempt to quantify the diversity of an author's vocabulary. Type token ratio (i.e. vocabulary size to the total number of tokens), the hapax legomena (i.e. words occurring once in a document), the hapax dislegomena (i.e. words occurring twice in a document) are some typical examples. The frequency of occurrence of context-free words (i.e. stop words) and the word or character n-gram approaches are some of the most important and effective methods in the field of authorship attribution.

In this paper, we conduct a comparative study of language modeling to traditional instance-based approaches. Our general approach is that we build a probabilistic language model for each author or train a classifier using as basic units of representation either characters, words, or part-of-speech (POS) tags. While POS tags have been used before in instance-based methods, e.g. [12], we first introduce them in this study as features in statistical language modeling for authorship attribution. We also experiment with linear combinations of several language models, as well as feeding unions of different feature types into instance-based methods. We are not aware of another study employing such combinations in the problem of authorship attribution.

Our experiments are conducted in a corpus consisting of 62,000 movie reviews written by 62 users. This corpus is selected because of its two interesting characteristics. First, it was edited by non-professional authors, rendering the task of authorship attribution more challenging and closer to the contemporary problem of authorship attribution on the Internet. Second, the numbers of documents and authors are both considered relatively large in comparison to traditional authorship attribution studies. Consequently, it would be interesting to see whether past results extend also to such large collections which are homogeneous in the sense that all authors treat the same topic.

2 Related Work

Previous work in authorship attribution studies focused mainly on the extraction of lexical features. Two are considered the state-of-the-art methodologies regarding the lexical approach: the multivariate vocabulary richness analysis and the frequency of occurrences of individual words [17].

Syntactic analysis has been less studied because of the limitations imposed by the language and the availability of a parser, a tool able to perform syntactic analysis of texts. However, in recent years there have been attempts to exploit syntactic information from texts. The idea behind this is that authors tend to unconsciously use specific syntactic patterns and particular sentence structures, which can be a reliable authorial fingerprint and facilitate authorship inference [18].

Baayen et al. [3] were the first to investigate the discriminatory potential of syntactic features for authorship attribution purposes. Based on a syntactically annotated corpus, which comprised around 20,000 words from two English books, they extracted rewrite rule frequencies. Their method outperformed the traditional word-based ones.

Stamatatos et al. [17] used Sentence and Chunk Boundaries Detector (SCBD), a robust and accurate NLP tool, to detect sentence and chunk boundaries in unrestricted Greek text. The text analysis was divided into three stylometric levels: token-level, phrase-level, and analysis-level. The two first levels consisted of measurements on sentence and phrase level respectively, such as length of noun phrases, length of verb phrases, noun phrase counts, and so on. The analysis level captured information omitted in the previous two levels regarding the way in which the input text was analyzed by the SCBD tool. Their method achieved 80% accuracy on 300 Greek texts written by 10 authors, containing a total of 333,744 words.

Recently, Pokou et al. [12] suggested the use of part-of-speech (POS) skip-grams in authorship attribution studies. Skip-grams are constructed like n-grams but allow a distance gap between adjacent POS tags. First, a set of training texts is pre-processed to become a sequence of POS tags, and a unique signature representing each author's style is extracted by using the most frequent part-of speech skip-grams in the training texts. Then, these signatures are used as a criterion to classify the test documents. In their experimentation, they used a collection of 30 texts, consisting of 2,615,856 words, written by 10 authors. Their method led to high classification accuracy.

Sidorov et al. in their recent work [16] introduced syntactic n-grams. Syntactic n-grams differentiate from classic n-grams because they take into account the position in which the elements are presented in the syntactic trees, not in the original text. Thus, they manage to capture syntactic relations between words. In the experiments conducted in a corpus of 39 documents written by 3 authors, Sidorov's method provided better results than the common n-gram approach for various n-gram lengths and types.

In this paper, we further investigate the use of syntactic information by building a separate language model for each author using part-of-speech tags as features to train the models. Our proposed method uses as features all the part-of-speech tags included in the training texts. No feature selection process to select the optimal number of features is required due to the fact that language models employ all features, and moreover, our set of features is already small consisting only of a handful of part-of-speech tags. For comparison, we also include in our experiments word and character level language models which were initially introduced by Fuchun Peng [11].

3 A Part-of-Speech Language Modeling Method

Our proposed method is based on the similarity-based paradigm. This approach includes the concatenation of all documents written by a certain author in a single profile, which is used for the extraction of the style-markers. For the evaluation process an attribution model is implemented to estimate the differences between every profile and an unseen text and the most likely author is chosen [18]. Specifically, we present a method for computer-assisted authorship attribution based on language models. This approach is composed of three phases described in the following subsections.

3.1 Preprocessing Phase

In this phase all the texts which are written by a certain author in the training corpus are concatenated in large files to create author profiles; thus, an author's profile is a union of all his considered known, or training, documents.

Using the Stanford NLP tagger [19], each word in author profiles is replaced by its corresponding part-of-speech tag from the Penn Treebank tagset [9]. Also, punctuation considered a useful literary style marker is preserved, because reviews are edited by non-professional authors who made wide use of punctuation. For example, consider the following text excerpt:

> *I bought Earthly Possessions because they filmed some scenes in New Jersey's Ocean Grove which is doubled as Perth, South Carolina. I really liked the chemistry between Sarandon and Dorff to surprise me.*

After the pre-processing step, it is transformed to a sequence of part-of-speech tags and punctuation marks:

> PRP VBD NNP NNP IN PRP VBD DT NNS IN NNP NNP POS NNP NNP WDT VBZ VBN IN NNP, NNP NNP. PRP RB VBD DT NN IN NNP CC NNP TO VB PRP.

In this step, every document of the test set is also pre-processed to obtain this form.

3.2 Language Modeling Phase

In this phase, every author profile created in the prepossessing phase is used to build a separate language model for each author. Next, we present the basic mathematical principles related to a statistical model of language.

Let us denote as $w_1^N = (w_1, w_2, ..., w_N)$ a sequence of N words. The probability of observing this text fragment under a language model can be computed as the conditional probability of every word in the fragment given the previous ones, i.e.

$$P(w_1^N) = \prod_{i=1}^{N} P(w_i | w_1^{i-1}) \,, \tag{1}$$

where $w_i^j = (w_i, w_{i+1}, .., w_{j-1}, w_j)$ is the sub-sequence from the i-th word to the j-th word, and w_i is the i-th word.

The above representation leads to a complex model because with a vocabulary size of V words, there are V^N possible sequences of the form $(w_1, w_2, ..., w_N)$. The above conditional probabilities constitute the free parameters of the model, which are learned from a training set. Obviously, even with reasonable magnitudes of V and N, we will never have enough training data to estimate V^N probabilities.

The need for a more simplified and compact model leads to the n-gram approach. In the n-gram model, without loss of generality, the probability of observing a new word is computed by taking into account only the previous $n-1$ words [1]. This approximation implies that the joint probability of the entire fragment can be calculated as

$$P(w_1^N) \approx \prod_{i=1}^{N} P(w_i | w_{i-(n-1)}^{i-1}) \,, \tag{2}$$

where $w_{i-(n-1)}^{i-1} = (w_{i-(n-1)}, \ldots, w_{i-1})$, and n the selected n-gram. Using this approximation, the number of the potential free parameters of a model with vocabulary size V are limited to V^n.

Let us define as count(w_i^j) the number of times the sub-sequence $w_i^j = (w_i, w_{i+1}, \ldots, w_j)$ appears in the training corpus. Then, the conditional probabilities of Eq. 2 can be estimated as

$$P(w_i | w_{i-(n-1)}^{i-1}) = \frac{\text{count}(w_{i-(n-1)}^{i})}{\text{count}(w_{i-(n-1)}^{i-1})}. \tag{3}$$

In practice, however, the probabilities in an n-gram model do not derive directly from the frequency counts, because it is likely for novel n-grams that were never explicitly witnessed in the training set to occur in the test set. Hence, a non-zero probability should be assigned to these unseen n-grams. There are many smoothing techniques used to confront this problem, including Good Turing discounting and back-off models. In this work we use the Witten-Bell discounting technique [6] because the size of the vocabulary is small containing 36 POS tags and 12 punctuation marks rendering smoothing not essential.

In this phase, every author profile created in the prepossessing phase, which consists of a sequence of tokens, is used to create a separate language model for each author by computing the aforementioned conditional probabilities (Eq. 3).

3.3 Authorship Attribution Phase

After the learning of models' parameters on the training corpus, we can classify unknown texts by how well each model predicts a text. For this purpose, we employ the Perplexity measure.

Given a test document $D = t_1^M = (t_1, t_2, ..., t_M)$, and considering an n-gram model, the intrinsic perplexity of the model on the test document is defined as:

$$\text{Perplexity}(D) = \sqrt[M]{\prod_{i=1}^{M} \frac{1}{P(t_i | t_{i-(n-1)}^{i-1})}}. \tag{4}$$

The lower a model's perplexity, the more likely the model is to predict the document. Thus, in the last phase, every unknown document of the test set is supplied to each language model, the perplexity of each model on the document is estimated, and the most likely author is selected.

4 Experimental Evaluation

In this section, we present the experimental results of comparing our proposed part-of-speech language model to traditional approaches. First, we describe the corpus and evaluation measures used.

4.1 Corpus and Evaluation Measures

We have experimented with the IMDB62 dataset which consists of 62,000 movie reviews written by 62 users, with exactly 1,000 reviews per user. The data were crawled from www.imdb.com by downloading all the reviews by prolific reviewers who submitted more than 500 reviews each. All downloaded texts belong to the period of May 2009 in order to minimize the risk of change of authorial style over time. We have chosen to use the IMDB62 dataset for the following three reasons:

- The data collected are homogeneous, because all texts deal with the same topic, making it more challenging to distinguish the stylistic idiosyncrasies of each author.
- The texts are written by regular people, not professional authors. In the traditional authorship attribution approaches, the training instances belong mostly to professional writers, whose style and language have been cultivated through the years, and as a consequence are distinguishable among the authors [4]. However, this dataset is edited by non-professional writers and the question that arises is whether authors with similar training and background are able to imprint their texts with their own unique authorial fingerprint.
- The number of candidate authors is relatively large compared to the majority of traditional authorship attribution studies.

Thus, this dataset is more challenging that others typically used in authorship attribution studies. While it was used at least once before for authorship attribution, e.g. [15], there is no extensive evaluation of different classifiers, feature types, and language models on this corpus, before our study. Others, e.g. [14], used this dataset for other tasks, such as sentiment classification.

For our evaluation, the corpus was divided randomly into a training and a test set. We selected 70% of the initial data set as a training set and the remaining 30% as a test set. We employed two evaluation metrics commonly used in text classification tasks: generalization Accuracy and macro-averaged F_1 measure.

4.2 Runs and Baselines

In order to set a baseline for the evaluation of the proposed method, we consider four types of features previously employed successfully in authorship attribution studies [5], namely: simple lexical (SL), simple character trigrams (CTG), content words (CW), and part-of-speech trigrams (POST).

The structure unit of the simple lexical features (SL) are the words. Based on the words, there have been extracted simple measurements that can be useful. The features extracted at token level include: the average and standard deviation of the words per sentence in each text, the total number of sentences in each text, the relative frequency of the tokens that are alphabetic units (calculated by dividing the total number of alphabetic elements of a text by the total number of tokens of every text), the relative frequency of the words longer that 15 letters, and the relative frequency of words shorter than 4 letters. Also, the features that represent the diversity of the vocabulary belong in the same set of features. In order to quantify the vocabulary's diversity of each author, the type-token ratio V/N (vocabulary size to the total number of words) was used. Two vocabulary functions count the so-called hapax legomena and hapax dislgomena, i.e. the number of the words of the text that appear only once or twice, respectively. We also implemented Yule's metric [21], Honore's metric [2], and the entropy function.

All the aforementioned measurements constitute the simple lexical features (SL) group. The second class of features consists of character trigrams (CTG). Previous research has shown that trigrams perform better than other character n-grams in an English corpus [8]. The third class of features includes words with the highest appearance frequency, the so-called content words (CW). Apart from features that rely on the words or n-grams of texts, the extraction of syntactic information is a reliable literary footprint [18]; thus, the fourth class of features includes part-of-speech trigrams (POST).

While SL features are arithmetic and small in number, CTG, CW, and POST, are tens of thousands or more. In order to reduce the computational load, we resort to feature selection. For the selection of features that carry significant information for classification, we implemented a feature selection method known to be among the best for text classification tasks [20], namely, the chi-square (χ^2) metric. Based on this metric, we selected the top-500 most informative features among the 10,000 most common features in the training corpus, per CTG, CW, and POST. The choice of the 500 cutoff comes from our preliminary experiments and is also supported by previous research, see e.g. [5,7]. Using more features did not seem to improve effectiveness, at an extra computational cost.

Each document of the training set is processed to produce a feature vector, a numerical vector consisting of the frequency of each feature of the feature set occurring in the document. Feature vectors are then used to train classifiers, via several algorithms, which are then applied to the test set to calculate generalization accuracy and macro-averaged F_1.

We paired each of the above four feature classes to four classification algorithms commonly and successfully used in authorship attribution studies [22]: Multinomial Naive Bayes (MNB), a Support Vector Machine (SVM), k Nearest Neighbour (kNN), and Random Forest (RF). For each of those classifiers parametrization is needed, however, parameter optimization is beyond the scope of our work. Hence, we used the default settings of scikit-learn[1], i.e. the machine learning Python library we employed for our experiments. As a baseline, we will select the best performing feature-class/classifier combination per evaluation measure; this constitutes a rather strong baseline.

Regarding language modeling, we experimented with using as features characters (C/LM), words (W/LM), and our proposed part-of-speech tags (POS/LM). For building and applying statistical language models, we employed the SRILM[2] toolkit comprised of a set of C++ classes. Again, while SRILM has some parameters, parameter optimization is beyond the scope of our work, so we used the default values (i.e. the default setting of SRILM for n-grams is 3). Note that, in contrast to instance-based methods where each text is represented as feature vector, as aforementioned, in language models there is no feature selection step; all features are used.

Table 1. Feature types, machine learning methods, and language models

SL	Lexical features based on simple measurements
CTG	500 most informative char 3-grams among the 10^4 most common 3-grams
CW	500 most informative words among the 10,000 most common words
POST	500 most informative POS 3-grams among the 10^4 most common 3-grams
MNB	Multinomial Naive Bayes with default settings (scikitlearn)
SVM	Support Vector Machines with linear kernel and default settings (scikitlearn)
kNN	k nearest neighbour with default settings (scikitlearn)
RF	Random Forest with default settings (scikitlearn)
C/LM	Language model with default settings (SRILM) and characters as features
UW/LM	Language model with context length 1 and words as features (SRILM)
W/LM	Language model with default settings (SRILM) and words as features
POS/LM	Language model with default settings (SRILM) and POS-tags as features

In summary, the feature types, machine learning methods, and language models we experimented with, are given in Table 1. Due to the settings described above, CTG, CW, and POST, are directly comparable to C/LM, UW/LM, and POS/LM, respectively, since they are using the same feature sets. These are character 3-grams, word unigrams, and POS 3-grams, respectively. W/LM is an

[1] www.scikit-learn.org.

[2] www.speech.sri.com/projects/srilm/.

extra run using word 3-grams. The language model based on unigrams (UW/LM) is not expected to perform since it is trivial.

Furthermore, we have tried combining feature types by (a) feeding unions of them into the classifiers, and (b) taking linear combinations of several language models with equal weights by simply adding their perplexities. A MinMax normalization process preceded the linear combination of language models because the ranges of perplexities produced by using different units seemed incomparable. This achieved better effectiveness in preliminary experiments (not reported here).

4.3 Results

In this section we present a set of experiments we ran in order to assess the performance of language modeling in comparison to the aforementioned baseline methods.

Table 2. Accuracy and F_1 on the test set of the IMDB62 corpus, for a variety of feature types and learning algorithms. Best results per feature type and per measure are in bold typeface; worst are with italics

Features/learner	Accuracy %				F_1 %			
	MNB	SVM	kNN	RF	MNB	SVM	kNN	RF
SL	14.3	**37.8**	*12.4*	31.5	12.8	**37.5**	*11.7*	30.8
CTG	82.7	**85.6**	*57.6*	*57.6*	82.9	**86.0**	58.2	*56.8*
CW	86.1	**88.6**	60.5	*58.6*	86.0	**89.0**	60.6	*57.8*
POST	53.5	**58.7**	34.0	*25.5*	52.4	**58.1**	33.4	*24.5*

Table 2 shows the generalization accuracy and macro-average F_1-measure of each combination of features and learning algorithms for the IMDB62 corpus. As it can be seen, the k-nearest neighbour (kNN) and Random Forest (RF) classifiers perform poorly on all feature sets for both evaluation metrics. Multinomial Naive Bayes (MNB) proves to be an effective learning method for almost all feature types except SL features, but Support Vector Machines (SVM) are superior to all other learning algorithms for all feature types.

Regarding the feature sets, simple lexical measures (SL) perform very poorly in all classifiers, so they do not seem to provide information relevant to the recognition of an author. While part-of-speech trigrams (POST) are better, character trigrams (CTG) and content words (CW) constitute more effective and reasonable choices of feature sets because they perform far better than the former two. The use of the syntax frequency tags (POST) fails to adequately describe the broader syntax structures and gather all the information about the syntax profile of each author. In summary, the best-performing feature-class/classifier combination is CW/SVM, in both evaluation measures, with CTG/SVM being very competitive. We will use both these runs as baselines.

Table 3. Accuracy and F_1 on the test set of the IMDB62 corpus, for a variety of combinations of feature types and learning algorithms. Best results per feature type and per measure are in bold typeface; worst are with italics

Features/learner	Accuracy %				F_1 %			
	MNB	SVM	kNN	RF	MNB	SVM	kNN	RF
CTG+CW	86.6	**91.2**	*61.5*	61.6	86.5	**91.2**	62.0	*60.9*
CTG+POST	84.4	**87.1**	59.3	*55.2*	84.2	**87.0**	59.8	*55.9*
CW+POST	85.3	**87.8**	59.3	*55.5*	85.2	**87.9**	59.8	*54.6*
CTG+CW+POST	87.4	**91.7**	63.0	*61.3*	87.3	**91.8**	63.5	*60.5*

Table 3 shows the results for all feature unions, except SL which were proven very weak above. Concerning the learning algorithms, we reach similar conclusions as above, i.e. SVM performs best, MNB following, and kNN, RF are the worst. Regarding feature combinations, we see that taking unions of features is generally beneficial to effectiveness: all combinations show improved performance than all the individual feature-types combined, except when CW combined with POST. This means that POST provide additional useful information in most cases. The union of all feature types (CTG+CW+POST) is the best run so far, closely followed by CTG+CW, both when fed into SVM. We will also use both these runs as baselines in order to compare the language models based on different units.

Table 4. Accuracy and F_1 on the test set of the IMDB62 corpus, for a variety of combinations of feature types and learning algorithms or language models. Best results per feature/learner class and per measure are in bold typeface

Features/learner	Accuracy %	F_1 %
CW/SVM	**88.6**	**89.0**
CTG/SVM	85.6	86.0
CTG+CW+POST/SVM	**91.7**	**91.8**
CTG+CW/SVM	91.2	91.2
C/LM	**92.3**	**92.7**
UW/LM	13.6	18.7
W/LM	84.4	85.2
POS/LM	89.5	89.8
C/LM+W/LM	93.6	93.8
C/LM+POS/LM	94.9	95.0
W/LM+POS/LM	94.1	94.3
C/LM+W/LM+POS/LM	**95.9**	**96.0**

Table 4 shows the language model runs (3rd batch of results) in comparison to the previously chosen baselines (1st and 2nd batches), as well several language model combinations. The trivial language model on unigrams (UW/LM) fails, as expected. From the rest, the pretty standard language model on word 3-grams (W/LM) is the weakest one, which performs slightly worse than the weakest of two baselines (CTG/SVM). The proposed language model on part-of-speech 3-grams (POS/LM) comes slightly above (rather insignificantly) the strongest baseline (CW/SVM), however, it has stronger efficiency benefits. The language model based on characters (C/LM) achieves a much higher performance.

Regarding the linear combinations of language models, all of them achieve better performance than the single-feature as well as the combined-feature baselines. Again here, we leave out of the combinations the very weak UW/LM. The combination of all the rest three language models achieves the best accuracy and F_1 of around 96%.

5 Conclusions

Traditional methods for automated authorship attribution employ several feature types and learning algorithms for building author profiles. Most previous research has dealt with small heterogeneous collections where each professional author may have been strongly associated with a topic. Furthermore, the style and language used by professional authors have been cultivated throughout the years, consequently becoming distinguishable, making authorship attribution relatively an easier task. We considered larger collections, with many non-professional authors, writing on a specific topic (homogeneous collection) such as movies. Our contributions are the following.

First, we evaluated the performance of four different feature classes commonly used in past literature, paired with four commonly used classifiers for the task. We found that Support Vector Machines paired with words or character trigrams as features are the most effective. This result is in-line with previous research, e.g. [8], so past results with instance-based methods seem to extend to larger homogeneous collections.

Second, we proposed a language model based on part-of-speech units and evaluated its performance against the former methods and other language models based on standard units such as characters or words. Here, in contrast to past literature, e.g. [10], where the word-level language model provides the best results, our experiment demonstrates that character or POS level language models achieve better classification results.

Third, we investigated combinations of features in learning algorithms by simply taking unions, as well as combining language models based on different units by taking a linear combination of their individual perplexity scores. Both combination methods seem to work well, achieving better results than the individual feature classes or language models they combine.

While our proposed POS/LM method provides only a slight effectiveness benefit over the best-performing standard methods, it has important efficiency benefits:

(a) building a language model on a handful of POS tags is fast, much faster than using characters or words as units, and (b) feature selection is not required in language models. Also, the attribution method of POS/LM avoids data sparsity problems, making smoothing non-essential. The vocabulary used for this model consists of 36 syntactic labels and 12 commonly used punctuation marks, eliminating the possibility for an unseen trigram of syntactic labels to arise in the test phase. There are no limitations imposed on the vocabulary, and every word in the English vocabulary, as well as novel word n-grams that were never witnessed in the training set, can appear in the test set.

Regarding the combination methods, while feeding unions of features into some classifier has no extra parameters, taking linear combinations of language models introduces some extra parameters: the coefficients of the linear combination. We have so far simply assumed equal weights by adding MinMax-normalized perplexities, nevertheless, this still achieved the best results in this paper. In this respect, optimizing in the future these coefficients could lead to even better effectiveness.

Acknowledgement. We thank Nektarios Mitakidis, master's student at our department, for his valuable guidance during the early stages of this work.

References

1. Allamanis, M., Sutton, C.: Mining source code repositories at massive scale using language modeling. In: Proceedings of the 10th Working Conference on Mining Software Repositories, pp. 207–216. MSR 2013. IEEE Press, Piscataway (2013)
2. Antony, H.: Some simple measures of richness of vocabulary. Assoc. Literary Linguist. Comput. Bull. **7**(2), 172–177 (1979)
3. Baayen, H., van Halteren, H., Tweedie, F.: Outside the cave of shadows: using syntactic annotation to enhance authorship attribution. Literary Linguist. Comput. **11**(3), 121–132 (1996)
4. Baayen, H., Halteren, H.V., Neijt, A., Tweedie, F.: An experiment in authorship attribution. In: 6th JADT I(January), pp. 69–75 (2002)
5. Grieve, J.: Quantitative authorship attribution: an evaluation of techniques. Literary Linguist. Comput. **22**(3), 251–270 (2007)
6. Ismail, R.: Comparison of modified kneser-ney and witten-bell smoothing techniques in statistical language model of bahasa Indonesia. In: 2nd International Conference on Information and Communication Technology (ICoICT), pp. 409–412, May 2014
7. Koppel, M., Schler, J.: Exploiting stylistic idiosyncrasies for authorship attribution. In: IJCAI 2003 Workshop on Computational Approaches to Style Analysis and Synthesis, pp. 69–72 (2003)
8. Koppel, M., Schler, J., Argamon, S.: Computational methods in authorship attribution. J. Am. Soc. Inf. Sci. Technol. **60**(1), 9–26 (2009)
9. Marcus, M., Kim, G., Marcinkiewicz, M.A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., Schasberger, B.: The penn treebank: annotating predicate argument structure. In: Proceedings of the Workshop on Human Language Technology, HLT 1994, pp. 114–119 (1994)

10. Peng, F., Schuurmans, D., Wang, S.: Augmenting Naive Bayes classifiers with statistical language models. Inf. Retrieval **7**(3), 317–345 (2004)
11. Peng, F., Schuurmans, D., Wang, S., Keselj, V.: Language independent authorship attribution using character level language models. In: Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics, EACL 2003, vol. 1, pp. 267–274. Association for Computational Linguistics, Stroudsburg (2003)
12. Pokou, Y.J.M., Fournier-Viger, P., Moghrabi, C.: Authorship attribution using variable length part-of-speech patterns. In: Proceedings of the 8th International Conference on Agents and Artificial Intelligence, pp. 354–361 (2016)
13. Raghavan, S., Kovashka, A., Mooney, R.: Authorship attribution using probabilistic context-free grammars. In: Proceedings of the ACL 2010 Conference Short Papers, ACLShort 2010, pp. 38–42 (2010)
14. Seroussi, Y., Zukerman, I., Bohnert, F.: Collaborative inference of sentiments from texts. In: Bra, P., Kobsa, A., Chin, D. (eds.) UMAP 2010. LNCS, vol. 6075, pp. 195–206. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13470-8_19
15. Seroussi, Y., Zukerman, I., Bohnert, F.: Authorship attribution with latent Dirichlet allocation. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning, pp. 181–189, CoNLL 2011. Association for Computational Linguistics, Stroudsburg (2011)
16. Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., Chanona-Hernández, L.: Syntactic dependency-based n-grams as classification features. In: Batyrshin, I., Mendoza, M.G. (eds.) MICAI 2012. LNCS (LNAI), vol. 7630, pp. 1–11. Springer, Heidelberg (2013). doi:10.1007/978-3-642-37798-3_1
17. Stamatatos, E., Fakotakis, N., Kokkinakis, G.: Computer-based authorship attribution without lexical measures. Comput. Humanit. **35**(2), 193–214 (2001)
18. Stamatatos, E.: A survey of modern authorship attribution methods. J. Am. Soc. Inf. Sci. Technol. **60**(3), 538–556 (2009)
19. Toutanova, K., Manning, C.D.: Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In: Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics, EMNLP 2000, vol. 13, pp. 63–70 (2000)
20. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: Proceedings of the Fourteenth International Conference on Machine Learning, ICML 1997, pp. 412–420. Morgan Kaufmann Publishers Inc., San Francisco (1997)
21. Yule, G.U.: The Statistical Study of Literary Vocabulary. Cambridge University Press, Cambridge (1944)
22. Zhao, Y., Zobel, J.: Effective and scalable authorship attribution using function words. In: Lee, G.G., Yamada, A., Meng, H., Myaeng, S.H. (eds.) AIRS 2005. LNCS, vol. 3689, pp. 174–189. Springer, Heidelberg (2005). doi:10.1007/11562382_14