Authorship Attribution of Internet Comments with Thousand Candidate Authors

Jurgita Kapočiūtė-Dzikienė, Andrius Utka, and Ligita Šarkutė

1 Vytautas Magnus University, K. Donelaičio 58, LT-44248, Kaunas, Lithuania
2 Kaunas University of Technology, K. Donelaicëio 73, LT-44029, Kaunas, Lithuania
jurgita.k.dz@gmail.com, a.utka@hmf.vdu.lt, ligita.sarkute@ktu.lt

Abstract. In this paper we report the first authorship attribution results for the Lithuanian language using Internet comments with a thousand of candidate authors. The task is complicated due to the following reasons: large number of candidate authors, extremely short non-normative texts, and problems associated with morphologically and vocabulary rich language.

The effectiveness of the proposed similarity-based method was investigated using lexical, morphological, and character features; as well as several dimensionality reduction techniques. Marginally the best results were obtained with the word-level character tetra-grams and entire feature set. However, the technique based on the randomized feature sets even using a few thousands of features achieved very similar performance levels, besides it outperformed method’s implementations based on the sophisticated feature ranking.

The best obtained f-score and accuracy values exceeded random and majority baselines by more than 10.9 percentage points.

Keywords: similarity-based paradigm, randomized feature set, internet comments, the Lithuanian language.

1 Introduction

The internet is replete with electronic text documents (comments, forum posts, tweets, etc.); however, the vast majority of them are written anonymously or pseudonymously. The anonymity factor makes the internet the perfect environment for expressing opinions on various issues. The internet is also the perfect place for negative deeds, e.g. harassment or bullying, impersonation of another individual, disclosure of the confidential information, and similar cyber crimes. The simple measures taken by the service providers trying to stop perpetrators are not always effective enough: he/she may change pseudonyms; write from the different IP addresses; or not reveal the real IP address by accessing websites through the Proxy servers. Even in these complicated cases when solely the plain text becomes the only evidence of an authorship, the identification problem can be still tackled due to an existing “stylometric fingerprint”: i.e. specific, individual, persistent, and uncontrolled habit of humans to express thoughts in the certain unique ways, of course, having in mind a scenario that author makes no efforts to modify his/her writing style. Hence, some anonymous writer may...
be prone to the specific expressions, vocabulary, emoticons, sentence structures, spelling errors or to have other recognizable idiosyncrasies. Van Halteren even named this phenomenon a “human stylome” [43] in deliberate analogy to the DNA “genome”. Despite strict implications not always seem to be absolutely correct, because “genome” is stable, but human writing style tends to evolve over time [11], “stylome” can still be added to human biometrics, next to voice, gait, keystroke dynamics, handwriting, etc.

The authorship analysis involves several main research directions: author verification, when deciding if a given text is written by a certain author or not (e.g. [20]); author profiling, when extracting information about the traits of an author: typically covering the basic demographic information that includes age (e.g. [33]), gender (e.g. [15]) or psychometric characteristics (e.g. [3]); plagiarism detection, when finding similarities between the texts (e.g. [40]); etc. However, in this paper we focus on the Authorship Attribution (AA) – a task of identifying whom, from a set of the candidate authors, is an actual author of a given anonymous text document. AA is one of the oldest Computational Linguistic problems, dating back to 1887 [24] which for a long time in the past was mainly restricted to the literary texts only. With the beginning of the internet era AA became highly topical and more focused on the wide range of practical problems, e.g. identification of offenders in anonymous harassment and threatening cases [42], tracking authors of malicious source code [2], etc. Therefore literary texts were replaced with e-mails [1], [44], web forum messages [37], online chats [5], [9], internet blogs [16] or tweets [34], [38], which, in turn, contributed to the development of Computational Linguistic methods able to cope effectively with these types of non-normative texts. Not only application domain is changing, a number of candidate authors is gradually increasing as well. If the first AA analysis attempts were able to cope with only a few or dozens of authors, recently it is applied on thousands candidate authors (e.g. 10,000 [16] and even 100,000 [27]) and limited training data, thus AA problem has become a so-called “needle-in-a-haystack” problem.

In this paper the AA problem is restricted to one thousand Lithuanian authors, using a corpus composed of the internet comments. However, our problem is complicated due to a number of reasons. Firstly, the used texts contain just a few words, but cover a wide range of topics. It is known that the shorter the text, the more difficult it is to determine its authorship; moreover, the writing style of an author may differ a bit depending on the topic he/she is discussing (especially in terms of vocabulary and expressions). Secondly, we have to deal with the Lithuanian language, which is rich in vocabulary and morphology, highly inflective, has a complex word derivation system, relatively free word-order in a sentence, and missing diacritics in non-normative texts. Thus, AA methods achieving a high accuracy for English do not necessary have to be the best for Lithuanian. Thirdly, the AA problem for Lithuanian has never been tackled using such a large number of candidate authors, therefore it is difficult to say in advance, if the methods solving AA problem for a few [12] or hundred candidate authors [14] can be still effective in this case.
2 Related Works

If excluding archaic rule-based approaches (attributing texts to authors depending on the rules, constructed by linguist-experts), automatic AA methods can be divided in two main paradigms: machine learning and similarity-based (for review see [39]). In the machine-learning paradigm, the texts of known authorship (training texts) are used to construct a classifier that can then be used to classify anonymous documents. In the similarity-based paradigm, an anonymous text is attributed to the particular author whose text is the most similar according to the calculated similarity measure. Broadly-speaking, the AA research is mainly focused on the choice of feature types for document representation, on the methods for a dimensionality reduction of feature space, and on either the choice of learning algorithms for the machine learning or the choice of similarity measures or distance metrics for the similarity-based approaches.

Since the first modern pioneering work (where Mosteller and Wallace [26] demonstrated promising AA results on The Federalist papers using Bayesian methods applied on the frequencies of a small set of function words) until 1990s AA was based on quantitative features (so-called “style markers”) such as a sentence or word length, syllables per word, type-token ratio, vocabulary richness functions, lexical repetition, etc. (the review of the early style markers can be found in [7]). However, these stylo-metric features are considered to be suitable only for the homogenous long texts (> 1,000 words) and for the datasets having small number of candidate authors.

In the contemporary research the most widespread approach is to represent text documents as vectors of frequencies, which elements cover specific layers of linguistic information (e.g. lexical, syntactic, semantic, character, etc.). Lexical features are the most commonly used feature type, which require only tokenization, moreover, they can be easy interpretable to humans. Lexical features can be divided into two types: function words (articles, conjunctions, prepositions, pronouns, etc., carrying no semantic information) and content words (sometimes used in a longer n-gram patterns) (relating author’s stylistic choices and providing topic information). When dealing with the multi-topic data the function words, which by consensus are considered as topic-neutral, are more often recommended [4] compared to the content words; however, function words still have to be used with caution, because some researchers proved them to correlate with the topic as much as with the authorship [25]. The effectiveness of syntactic and semantic features usually depend on the accuracy of applied linguistic tools (e.g. part-of-speech taggers, parsers) or exhaustiveness of external data resources (e.g. thesauruses, databases, ontologies). Although used alone syntactic or semantic features can hardly outperform lexical, but often show improvements when applied in combination [6]. However, character features are considered to be the most important document representation type for author’s style detection: they are language-independent, able to capture style through lexical and contextual information, are tolerant to grammatical or typing errors, and able to handle limited data (the robustness of character n-grams was proved by many researchers, e.g. in [21], [41]).

After extracting the features, the most popular attempt to increase AA accuracy is based on the assumption that irrelevant and noisy features should be omitted. It is
usually done by applying wrapper (performing a search over all possible subsets of features) or filtering (scoring the features first and then omitting the least informative ones) features selection methods on the primary feature set before the attribution process (the review about feature selection methods is in [8]).

AA task is the most often solved using machine learning methods (for the review see [35]). In the contemporary computational research, Support Vector Machines (SVMs) (considered as the most accurate thus the most suitable technique for text classification) are the most popular choice for AA tasks; however, other methods are explored as well. Some comparative experiments with Decision Trees (DTs), Back Propagation Neural Networks (BPNNs) and SVMs revealed that SVMs and BPNNs achieve significantly better performance compared to DTs [46]. However, comparative experiments prove that similarity-based methods often outperform machine learning techniques; e.g. Memory-Based Learning (MBL) produces better results compared to Naive Bayes (NB) and DTs [45]; the Delta method can surpass the performance levels achieved by the popular SVMs [10]. Different proposed improvements for the similarity-based approaches contribute to their effectiveness even more. Thus, novel classification scheme based on the specific vocabulary outperforms Principal Component Analysis (PCA) and Delta [31]; Latent Dirichlet Allocation (LDA)-based classification scheme surpasses two classical AA approaches based on the Delta rule and chi-squared distance [32].

Virtually all previously discussed research works have focused on the problems with a small number of candidate authors and only recently larger datasets (in terms of a number of those authors) have been considered. Despite all advantages of the machine learning techniques, similarity-based approaches are considered to be more suitable for the problems with more candidate authors and limited training data (e.g. MBL method applied on 145 authors outperformed SVMs [22]; MBL method applied on 100,000 authors outperformed NB, SVM and Regularized Least Squares Classification (RLSC) [27]). However, some researchers argue that weaknesses of machine learning approaches are due to the difficulties when configuring complex models with many parameters having only a small number of training instances and, in particular, offer to use normalization which can make the huge impact: e.g. after normalization RLSC performs equally well as MBL [27]. Besides the same researchers experimenting with a blog corpus of 100,000 candidate authors developed a novel technique to estimate a confidence of the classifier outputs (by measuring a difference between the best and second-best matching classes, running two different classifiers and outputting the result only if they agree, and combining the results by the meta-learning) which resulted in increased overall precision. In [34] researchers experiment with the Twitter corpus (containing very short texts, limited to 140 characters) of at most 1,000 candidate authors, use SVMs machine learning classifier and demonstrate that introduced AA feature type (so-called “flexible patterns”, capturing the context in which function words are used) is achieving significant improvement over baselines based on the character or word n-grams. Some researchers [36] argue that AA problem dealing with a blog corpus of over 19,000 candidate authors can be effectively solved with the similarity-based LDA (in particular, generative probabilistic model, where each text document is generated according to the distribution of topics and each word in
the document – according per-topic word distribution) by calculating the distance between the LDA-based representations in the training documents of known authorship and anonymous text document. The researchers claim that LDA approach applied on the author profiles (created from concatenated training text documents) yields state-of-the-art performance in terms of accuracy with enough training data. The hybrid method dealing with three text representations types (tf-idf restricted to content words; binary idf restricted to content words; tf-idf on different stylistic features) and combining similarity-based with machine learning approaches, described in [18], effectively copes with 10,000 blog authors. Firstly, researchers rank authors by cosine similarity using all three text representation techniques, if at least one of them gives the top-rank, the pair (constructed of an anonymous text and the author ranked most similar for that text) is tested on the meta-learning SVM classifier: in case of success the anonymous text is attributed to that author, otherwise the method outputs an answer “don’t know”. The similarity-based approach using cosine measure applied on the blog dataset containing 10,000 candidate authors and character tetragrams along with the multiple randomized feature sets can achieve high precision [16, 17, 19]. The researchers also adjusted the method to cope with the open-class cases: if calculated score value (aggregating several attribution decisions) is below some determined threshold the anonymous text is considered to belong none of the candidate authors. Nevertheless, the methods dealing with a huge number of the candidate authors are rather slow. In [28] researchers experiment with the corpus containing Japanese microblogs written by 10,000 candidate authors and test similarity-based methods using cosine measure and character n-grams (where n is 1, 2, and 3). The proposed novel weighting scheme (which strengthens the weight of n-gram depending on its n) applied on the morphemes converted to the part-of-speech-tags significantly shortens attribution execution time.

Considering all previously surveyed methods and given recommendations, we may conclude that similarity-based paradigm should be better choice for our task. However, the feature representation type and the feature selection method can be chosen only after experimental investigation.

3 Methodology

3.1 The Corpus

We composed our corpus of the internet comments harvested in January 2015 from the Lithuanian news portal www.delfi.lt. These comments were posted by anonymous users expressing their opinions about articles in two subjects “In Lithuania” and “Abroad”. All text fragments containing non-Lithuanian alphabet letters (except punctuation marks and digits) were eliminated; all the replies and meta-information were filtered out as well leaving just the plain texts. Besides, the texts shorter than 30 symbols (excluding white-space characters) were not included into the corpus.

The composed corpus contains 1,000 authors. We made an assumption that identity of author can be revealed, if his/her texts are written under the same unique IP address and the same unique pseudonym (taking both together as a single unit). Although
some exceptions (when a few authors write under the same pseudonym using the same IP address) may occur, these noisy cases are rather rare to make a significant influence on the overall AA results. However, in the real world applications ideal conditions are hardly possible; but on the other hand, the real environment is exactly what is needed to compose a bold and undistorted testbed for AA task: internet comments cover wide range of topics, are single units (not only the extracted snippets of long texts as it is done in e.g. [16] or [30]), do not necessary refer to each other; moreover, authors, hiding behind the anonymity curtain have no reasons to pretend “better” and therefore do not make any efforts to refine their writing style. As the result of it, the texts are full of out-of-vocabulary words, including diminutives and words with missing diacritics (where Lithuanian letters are replaced with appropriate Latin letters, e.g. ė → e, š → s, ū → u, etc.).

The useful statistics about the composed corpus is given in Table 1. Random $\sum_{j} P^2(c_j)$ and majority $\max\{P(c_j)\}$ baselines (where $P(c_j)$ is the probability of some author $c_j$: texts written by particular $c_j$ are divided by all texts in the corpus) show the lowest accuracy threshold which must be exceeded to claim that applied method is effective and reasonable enough for our AA task. Although on average one author gets ~35 texts, the distribution is not balanced. The smallest number of texts per author is 12, the largest – 425. Almost half of the authors (in particular, 554) have produced less than 25 texts and only 45 authors more than 100 texts.

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of authors</td>
<td>1,000</td>
</tr>
<tr>
<td>Number of texts</td>
<td>34,946</td>
</tr>
<tr>
<td>Number of tokens (words/digits)</td>
<td>940,022</td>
</tr>
<tr>
<td>Avg. text length in tokens</td>
<td>26.899</td>
</tr>
<tr>
<td><strong>Random baseline</strong></td>
<td>0.002</td>
</tr>
<tr>
<td><strong>Majority baseline</strong></td>
<td>0.012</td>
</tr>
</tbody>
</table>

### 3.2 Formal Description of the Task

In essence, the AA task which we are solving in this paper can be formally described as follows:

The corpus $D$ contains text documents $d_i$ attributed to a closed-set of candidate authors (defined as classes) $C = \{c_i\}$. Any author’s $c_j$ profile $Pr_j$ can be created by concatenating all $d_i$ attributed to $c_j$ into a single text document.

According to the determined sequence of features $F_k$ (distinct text elements, e.g., tokens, lemmas, etc.) each $d_i$ or $Pr_j$ can be transformed into appropriate vectors $\mathbf{v}_i = \langle v_m \rangle$ or $\mathbf{v}_j = \langle v_m \rangle$, respectively, with elements representing absolute counts of these features found in the documents.

Function $\phi$ determines the mapping, relating text documents $d_i$ with their authors $c_j$, i.e. $\phi : D \rightarrow C$. Our goal is to offer a method, which could find as close approximation of $\phi$ as possible.
3.3 Similarity-Based Method

A decision to use the similarity-based paradigm for our solving task was already taken after the comprehensive analysis of related works (see Section 2); and a concept of the randomized feature set for our AA method was borrowed from [16, 17], [19].

Repeat $K$ times:
- Compose feature set $F_k$
- Create $\vec{v}$ for $d$ depending on $F_k$
  - For each $j^{th}$ author:
    - Construct profile $Pr_j$ by concatenating $d_i \in D$
    - Create $\vec{v}_j$ for $Pr_j$ depending on $F_k$
      - Memorize calculated $\text{sim}(\vec{v}, \vec{v}_j)$: $\text{SIM} = \text{SIM} \cup \text{sim}(\vec{v}, \vec{v}_j)$
      - If $\text{max}((\text{SIM})) > \sigma_1$:
        - Memorize determined candidate author $c_j$: $\text{CA} = \text{CA} \cup c_j$
  - For each unique $c_j \in \text{CA}$:
    - Calculate a number of times $c_j$ is in $\text{CA}$: $\text{times}(c_j \in \text{CA})$
    - Memorize proportion: $\text{SCORE} = \text{SCORE} \cup \text{times}(c_j \in \text{CA}) / K$
    - If $\text{max}((\text{SCORE})) > \sigma_2$:
      - Anonymous text $d$ is attributed to the determined $c_j$

In our method we are calculating a similarity between anonymous text document $d$ and all profiles of the authors $Pr_j$. Indeed, profile-based approaches have advantages over instance-based when the text documents are very concise (as it is in our case, where an average text length is only ~27 tokens, see Table 1), then concatenation helps to create sufficiently long document, which is already suitable for capturing the author’s writing style.

For calculating the similarities between the vectors $\vec{v}$ and $\vec{v}_j$ we have chosen one of the most popular similarity measures – usual cosine similarity [29] (used by many researchers for their AA tasks even without any considerations), presented in eq. 1. Calculated cosine similarity values fall into an interval ranging from 0 to 1, where 0 indicates that the vectors are not similar, 1 – that they are equal.

$$\text{sim}(\vec{v}, \vec{v}_j) = \frac{\sum_{n=1}^{N} v_{n} \times v_{n,j}}{\sqrt{\sum_{n=1}^{N} (v_{n})^2 \times \sum_{n=1}^{N} (v_{n,j})^2}}$$  (1)

Any other measure can be selected instead; however its effectiveness should be investigated experimentally. For the comparison reasons we investigated Euclidean distance metric. Since obtained results gave no statistically significant improvements over cosine similarity, they are not presented in this paper.
The feature types; the feature set $F_i$ size $N$; the values of two thresholds $\sigma_1$ and $\sigma_2$ still have to be investigated experimentally.

### 3.4 Main Research Directions

Our main focus is on the following research directions, which should fill the gaps in the AA method, described in Section 3.3:

1. Feature type for the text documents representation. We explored the impact of the most popular feature types, covering lexical, morphological, and character levels (for the statistics see Table 2):

   - **lex** – tokens (words + digits). It is the most popular lexical feature type (in our case involving both content-specific information and function words). This feature type is one of the most popular types used in AA.
   - **lem** – lemmas, based on the word tokens. To obtain lemmas (canonical forms of words) the texts were processed with the Lithuanian morphological analyzer-lemmatizer “Lemuoklis” [47], which changes recognized words into their lemma, transforms common words into the lower case and replaces digits with the special tag. “Lemuoklis” is not adjusted to deal with the out-of-vocabulary words, therefore it could not recognize even $\sim$19.16\% (or 180,139 words) of all words in the corpus, leaving them in the original untouched form. This feature type is especially recommended for the morphologically rich languages.
   - **fwd** – function words. The content-free lexical feature which includes prepositions, pronouns, conjunctions, particles, interjections, and onomatopoeias. The words of these part-of-speeches were recognized by “Lemuoklis”, instead of using pre-established list of function words. This feature type by consensus is considered as topic-neutral and by many researchers was proved to be a relatively good identifier of the author’s writing style.
   - **chr4** – word level character tetra-grams. This character feature type is based on successions of 4 characters within the token boundaries (a window is sliding one character at the time). E.g. **chr4** for phrase “authorship attribution” would produce the following word-level character n-grams: “auth”, “utho”, “thor”, “hors”, “orsh”, “rshe”, “ship”, “atth”, “trth”, “trib”, “ribu”, “ibut”, “buti”, “utio”, and “ition”. We selected tetra-grams, because this feature type was already proved to be the best choice on the Lithuanian texts for the topic classification among all other n-grams and even other feature types, based on more sophisticated lexical and morphological information [13].

Besides, we also explored the impact of the diacritics-free character tetra-grams. Since an optional usage of diacritics makes the data very sparse (which may result in the lower accuracy) and there are no effective tools able to restore all missing diacritics, we replaced Lithuanian letters with the appropriate Latin letters before applying tetra-gram tokenizer.
Table 2. Statistics about the features in the corpus

<table>
<thead>
<tr>
<th>Feature type</th>
<th>Maximum number of features</th>
</tr>
</thead>
<tbody>
<tr>
<td>lex</td>
<td>137,748</td>
</tr>
<tr>
<td>lem</td>
<td>83,228</td>
</tr>
<tr>
<td>fwd</td>
<td>1,750</td>
</tr>
<tr>
<td>chr4</td>
<td>65,746</td>
</tr>
<tr>
<td>diacritics-free chr4</td>
<td>52,286</td>
</tr>
</tbody>
</table>

2. The dimensionality reduction of the feature set, when all irrelevant and noisy features are eliminated, is considered one of the key factors helping to increase the accuracy. We explored the impact of the following dimensionality reduction techniques:

- **No reduction**, i.e. all available features (see in Table 2).
- **Ranking and selecting the best N features**. The features were ranked according to the calculated chi-squared values (the ranking was done with ChiSquaredAttributeEval function implemented in Weka 3.7 Machine Learning toolkit)\(^2\), and only then the top N features were selected to form \(F_k\). A number of N was varied in our experiments having 1,000; 5,000; 10,000; 20,000; 30,000; 40,000; and 50,000.
- **Random selection of N features**. N features were randomly selected from the entire composed feature set. We explored the same N values as described in the previous item. This dimensionality reduction technique was used with the randomized feature sets only, where the final attribution decision was taken after aggregation and generalization of all decisions obtained in \(K > 1\) iterations (see Pseudo-code in Section 3.3).

3. The influence of thresholds:

- **Threshold \(\sigma_1\)** indicates the lowest cosine similarity value necessary to determine that the compared vectors are similar enough. We investigated the values from 0 to 1 with the interval of 0.1.
- **Threshold \(\sigma_2\)** indicates the lowest score value necessary to attribute anonymous text document to the particular class. This threshold is used with the randomized feature sets only having \(K > 1\). We investigated the values of \(\sigma_2\) equal to 0.1, 0.2, 0.4, 0.6, 0.8 and 1.0.

The default values in the preliminary experiments for \(K\), \(\sigma_1\), and \(\sigma_2\) are 1, 0, and 0.1, respectively.

---

\(^2\) Downloaded from http://www.cs.waikato.ac.nz/ml/weka/.
4 Experiments and Results

Our experiments involved several research directions, presented in Section 3.4.

During a single experiment we performed 50 runs, testing 100 text documents in each run; hence, the method (presented in Section 3.3) was tested 5,000 times. During a single test text document was randomly selected from $D$, making an assumption that it’s class is not known in advance; besides selected document was automatically removed from the dataset $D : D = D \setminus d$. The class of $d$, attributed by the method had to be compared with its real class. Despite our corpus is not balanced, a randomization function (used for selecting “anonymous” $d$) was not biased towards the largest classes (having the most texts), because the author was randomly selected at first, but his/her text was selected only afterwards.

The evaluation metrics— in particular, accuracy (presented in eq. 2) and $f$-score (presented in eq. 3) were calculated for each run (testing 100 attributed text documents) and averaged in 50 runs.

$$
\text{accuracy} = \frac{tp + tn}{tp + tn + fp + fn}
$$

where $tp$ – true positives ($c_j$ texts correctly attributed to $c_j$); $tn$ – true negatives (any other $c_j'$ correctly attributed to $c_j'$); $fp$ – false positives ($c_j'$ erroneously attributed to $c_j$), $fn$ – false negatives ($c_j$ erroneously attributed to $c_j'$).

$$
\text{f\_score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
$$

where precision and recall are presented in eq. 4 and eq. 5, respectively.

$$
\text{precision} = \frac{tp}{tp + fp}
$$

$$
\text{recall} = \frac{tp}{tp + fn}
$$

To make sure that the differences between the obtained results are statistically significant we performed McNemar test [23] at the significance level $\alpha = 0.05$, meaning that the differences are considered statistically significant if calculated probability density function $P < \alpha$.

During a preliminary experiment we experimentally investigated the influence of the feature representation type, using no dimensionality reduction (see Fig. 1). The differences in accuracies between the best determined feature type chr4 and all the rest types lex, lem, fwd are statistically significant with $P$ values equal to 0.006, 0.016, 0.000, respectively. We also ran control experiment using diacritics-free chr4 and obtained averaged $f$-score and accuracy values equal to 0.105 and 0.119, respectively.
Despite these values are a bit lower compared to chr4, the difference in accuracy is not statistically significant, because $P = 0.386 >> \alpha$.

![Fig. 1. The influence of the feature type on the results. The averaged values are given in columns (gray and white for f-score and accuracy, respectively) with the confidence intervals above; the black curve indicates the higher of random or majority baselines.](image)

Using the best determined feature type chr4 and no dimensionality reduction we ran another control experiment testing the influence of $\sigma_1$ value on the results. Starting from 0 we gradually increased $\sigma_1$ by 0.1 until it reached 1. The average f-score/accuracy values were gradually deteriorating and approaching to 0: 0.121/0.135, 0.106/0.117, 0.035/0.037, 0.009/0.009, 0.003/0.003, 0.000/0.000.

Using the best feature type chr4 and $\sigma_1=0$ we explored the influence of the dimensionality reduction in terms of $N$; feature ranking; and the effect of the randomized feature set with $K=20$ (see Fig. 2). Since the results obtained with $N = 30,000$ and $N = 50,000$ show almost the same trend as $N = 40,000$, we do not present them in Fig. 2 not to overload the reader with too much information. The differences in accuracies between the best obtained results (with all the features) and randomized feature set when $N$ is 5,000, 10,000, 20,000, 30,000, 40,000, 50,000 are not statistically significant with $P$ values equal to 0.039, 0.676, 0.940, 0.764, 0.949, 0.974, respectively; but significant when $N = 1,000$ with $P = 0.000$. The differences between the randomized feature set and appropriate top-ranked feature set are statistically significant when $N$ is 1,000, 5,000, 10,000, and 20,000 with $P$ values equal to 0.672, 0.001, 0.000, 0.037, respectively; and not statistically significant when $N$ is 30,000, 40,000, 50,000 with $P$ values equal to 0.983, 0.966, 0.966, respectively.
Fig. 2. The influence of $N$ on the evaluation metrics. The first two columns next to each $N$ represent the results with the top-rank feature set, the second two – with the randomized feature sets and $K = 20$. For the other notations see the caption of Fig. 1.

Using $chr4$ and $\sigma_1=0$ we ran another control experiment to test the influence of $\sigma_2$ value on the results. The best results were achieved with the default 0.1 and any increase gave a drop in $f$-score and accuracy. However, a decline with the higher $N$ values was much steeper compared with the lower, e.g. with $N = 50,000$ and $\sigma_2$ equal 0.1, 0.2, 0.4, 0.6, 0.8, and 0.9 we obtained $f$-score/accuracy values equal to 0.120/0.135, 0.120/0.135, 0.118/0.131, 0.106/0.116, 0.088/0.096, 0.061/0.066, respectively.

5 Discussion

All experimentally obtained results presented in Fig. 1 and Fig. 2 are reasonable and effective enough, because exceed random and majority baselines.

The best determined feature type for the text documents representation is word-level character tetra-grams (giving the maximum $f$-score and accuracy of 0.121 and 0.135, respectively, with all available features). Neither tokens, nor token lemmas could outperform character tetra-grams. This probably happened due to the larger vocabulary, where all different inflective word forms have to be considered as the different features (making feature set very sparse), thus it is rather hard to find their matches between the anonymous text and profiles of the authors. Since we were dealing with the internet comments, lemmatizer could not recognize out-of-vocabulary words, therefore token lemmas, which are considered to be the most accurate type for the morphologically rich languages appeared only in the second best place. If tokens and token lemmas are too sparse to find their matches, function words are too rare to find consistent patterns between anonymous text and the profiles. It is likely that
function words would produce better performance in the instance-based attribution scenario.

If compared character tetra-grams with diacritics-free character tetra-grams, simple tetra-grams are marginally better. It seems that the author’s choice to use or not to use diacritics in the texts is rather important constituent of his/her writing style.

Since attributed texts are very short (average length is only ~27 tokens), even a poor similarity (in terms of $\sigma_1$ values) detected between it and the author’s profile already makes positive impact on the results.

Marginally the best results were achieved with the entire feature set; however, randomized feature sets containing only 5,000 features reached very similar AA performance levels (due to the statistical significance between the differences of the obtained results). The technique based on the top-ranked $N$ features is not as effective as randomized feature sets: it could reach the same performance level only with $N \geq 30,000$. Thus, the power of aggregated decision is much stronger compared to the single one, despite that one is based on the top-ranked features.

Of all tested $\sigma_1$ values the best AA results were obtained with 0.1, which means if at least in 2 of 20 iterations some anonymous text is attributed to the same class it is already enough to make a final attribution decision. Thus, dealing with such sparse data and such short texts even disputed decision already seems to be better choice than no decision at all.

Unfortunately obtained results cannot be directly compared with the results for the other languages due to the very different experimental conditions: number of candidate authors, language types, text length, dataset sizes, etc.

6 Conclusions and Future Work

In this paper we were solving authorship attribution problem for one thousand candidate authors using Lithuanian internet comments. To tackle this problem we experimentally investigated the proposed similarity-based approach, the effects of the feature representation types (lexical, morphological, character), and several dimensionality reduction techniques (no reduction, top-ranked feature selection, randomized feature sets) on the attribution results.

Since the texts are too complicated to be effectively processed with the external morphological tools, the best results were obtained using word-level character tetra-grams: the baselines were exceeded by more than ~10.9% achieving 12.1%/13.5% of $f$-score/accuracy.

Marginally the best results were achieved with no feature reduction, however randomized feature sets with only a few thousands of features reached very similar performance levels. The technique based on the randomized feature sets outperformed the implementations based on the feature ranking.

In the future research we are planning to perform exhaustive linguistic error analysis, which could give us ideas on how to improve the method further; to investigate different similarity measures, and to test the method on the expanded number of the candidate authors up to tens of thousands.
Acknowledgments

Research was funded by a grant (No. LIT-8-69) from the Research Council of Lithuania.

References

44. de Vel, O., Anderson, A.M., Corney, M.W., Mohay, G.M.: Mining e-Mail Content for Author Identification Forensics. SIGMOD Record 30(4), 55–64 (2001)