Çankaya Üniversitesi Fen-Edebiyat Fakültesi,
Journal of Arts and Sciences Sayı: 7, Mayıs 2007

Author Identification for Turkish Texts

Tufan TAŞ[1], Abdul Kadir GÖRÜR[2]

The main concern of author identification is to define an appropriate characterization of documents that captures the writing style of authors. The most important approaches to computer-based author identification are exclusively based on lexical measures. In this paper we presented a fully automated approach to the identification of the authorship of unrestricted text by adapting a set of style markers to the analysis of the text. In this study, 35 style markers were applied to each author. By using our method, the author of a text can be identified by using the style markers that characterize a group of authors. The author group consists of 20 different writers. Author features including style markers were derived together with different machine learning algorithms. By using our method we have obtained a success rate of 80% in avarege.

Introduction

There exist very different types of documents on all over the Internet. Published texts on the Internet provide the ability to process them by using some special software. The motivation behind this software is the need for rapid retrieval of the required data, search for specific information and some language specific techniques such as Natural Language Processing (NLP) [12]. Natural Language Processing is a research area that is used for many different purposes and it becomes more popular continuously. Speech syntheses, speech recognition, machine translation, spelling correction are some of the application of NLP.

[1] Computer Engineering Department, Çankaya Univeristy, Anakara
e-mail: tufantas@gmail.com
[2] Computer Engineering Department, Çankaya Univeristy, Anakara
e-mail: agorur@cankaya.edu.tr

Individuals have distinctive ways of speaking and writing, and there exists a long history of linguistic and stylistic investigation into author identification. In recent years, practical applications for author identification have grown in areas such as intelligence, criminal law, civil law, and computer security. This activity is part of a broader growth within computer science of identification technologies, including, cryptographic signatures, intrusion detection systems, and others. Automating author identification [2, 8, 9, 11, 16] promises more accurate results and objective measures of reliability, both of which are critical for legal and security applications. Recent research has used techniques from machine learning [4, 5, 14] and natural language processing author identification.

Author identification is the task of identifying the author of a given text. It can be considered as a typical classification problem, where a set of documents with known authors are used for training and the aim is to automatically determine the corresponding author of an anonymous text. In contrast to other classification tasks, it is not clear which features of a text should be used to classify an author. Consequently, the main concern of computer-based author identification is to define an appropriate characterization of documents that captures the writing style [20] of authors.

Author identification has a long history that includes some famous disputed authorship cases and also has forensic applications. The advent of non-traditional author identification techniques can be traced back to 1887, when Mendenhall [21] first created the idea of counting features such as word length. His work was followed by work from Yule and Morton [17] with the use of sentence lengths to judge authorship. Brainerd [17] concentrated on syllables per word. Moreover, Holmes [7] developed a function to relate the frequency of used words and the text length. Karlgren-Cutting [15] figured out the style marker of the text. Biber [6] added the syntactic and lexical style markers. In the recent improvements on author identification we can see Kessler [3], who developed a fairly simple and reliable method. Twedie and Baayen [10] showed that the proportion of the different word count to the total word count could be a fair measurement and the results for the texts which are shorter than 1000 word in length could be inconsistent. Burrows [13] used principal components analysis (PCA) to find combinations of style markers that can discriminate between a particular pair (or small set) of authors. Another related class of techniques that have been applied are machine learning algorithms which can construct discrimination models over large numbers of documents and features. Such techniques have been applied widely in topic-based text categorization [1] and other stylistic discrimination tasks, as well as for author identification. Such techniques have been applied widely in topic-based text

categorization and other stylistic discrimination tasks. Often, studies have relied on intuitive evaluation of results, based on visual inspection of scatter plots and cluster analysis trees, though recent work has begun to apply somewhat more rigorous tests of statistical significance and cross validation accuracy. Other stylometric features that have been applied include various measures of vocabulary richness and lexical repetition, based on Zipf's [18] studies on word frequency distributions. Most such measures, however, are strongly dependent on the length of the text being studied, and so are difficult to apply reliably. Many other types of features have been applied, including word class frequencies, syntactic analysis, word collocations, grammatical errors, and word, sentence, clause, and paragraph lengths. Many studies combine features of different types using multivariate analysis techniques.

Author identification can be used in a broad range of applications, to analyze anonymous or disputed documents/books. In Plagiarism detection which can be used to establish whether claimed authorship is valid. In criminal investigation as Ted Kaczynski [19] was targeted as a primary suspect in the Unabomber case, because author identification methods determined that he could have written the Unabomber's manifesto. In forensic investigations where verifying the authorship of e-mails and newsgroup messages, or identifying the source of a piece of intelligence.

This research has attempted, to determine if there are objective differences articles from different authors. To determine if an author's style is consistent within their own texts. To determine some method to automate the process of authorship identification.

The rest of the paper is organized as follows. Methods / Experiments include detailed description of the methods and techniques used in the project. Results and Discussions include the experimental results of the project which is clearly stated and discussed. Summary depicts our conclusions.

Methods / Experiments

For the author identification system a corpus has been developed in which there are texts from different newspapers. The subjects of these texts range from current events to medical and political issues. The author-based corpus contains two sets: test and training. The training set contains 20 different texts for each one of 20 different authors. On the other hand, for the test set there are 5 different texts for each one of 20 different authors. An author may be focused on a specific subject in certain time intervals and it may affect the features which are derived from articles of that author. Therefore articles are not selected consecutively.

After we had established the corpus, we studied style markers. We used 35 style markers which are shown in Table 1. These 35 style markers have been processed for every article of each author. We were able to collect 35 style markers per author with an average of 20 articles for each author in the training set. Style markers determine the number of words and sentences between SM1 and SM7, the word types between SM8 and SM21, the number of punctuation marks between SM22 and SM27, and word based features between SM28 and SM35.

Code	**Style markers**	**Code**	**Style markers**
SM1	# of sentences	SM19	Average # of question words
SM2	# of words	SM20	Average # of reflections
SM3	Average # of words	SM21	Average # of cursors
SM4	Average word length	SM22	# of point
SM5	Average # of short words	SM23	# of comas
SM6	# of different words	SM24	# of colons
SM7	Word richness	SM25	# of semicolons marks
SM8	Average # of nouns	SM26	# of question marks
SM9	Average # of verbs	SM27	# of exclamation marks
SM10	Average # of adjectives	SM28	Guirad's R
SM11	Average # of adverb	SM29	Herdan's C
SM12	Average # of particle	SM30	Rubet's K
SM13	Average # of pronoun	SM31	Maas' A
SM14	Average # of conjunctions	SM32	Dugasts U
SM15	Average # of exclamations	SM33	L. Janenkov And Neistoj Measure
SM16	Average # of proper names	SM34	Brunet's W
SM17	Average # of numbers	SM35	Sichel's S
SM18	Average # of abbreviations		

Table 1 Style Markers

Guirad's $R = V/\sqrt{N}$
Herdan's $C = \log_{10}V / \log_{10}N$
Rubet's $K = \log_{10}V / \log_{10}(\log_{10}N)$
Maas' $A = \sqrt{(\log_{10}N / \log_{10}V) / (\log_{10}N)^2}$
Dugasts $U = (\log_{10}N)^2 / (\log_{10}N / \log_{10}V)$
L. Janenkov And Neistoj Measure $= 1 / (V^2 \times \log_{10}N)$
Brunet's $W = N^{V / 0.172}$
Sichel's S = count of hapax dislegomena / V

The number of different words (types) and the total number of words (tokens) can be counted to calculate a type-token ratio. The number of words used once

(hapax legomena) or twice (hapax dislegomena) can be counted. A set of metrics based on the values of types and tokens in a document were also used as word-based features. Style markers from SM28 to SM35 are explained above, where N is the total number of tokens, V is the total number of types and count of hapax dislegomena is defined as the number of types that occur twice in the text.

After working on the corpus and style markers, we indexed our corpus by giving a document ID to each document, author ID to each author and index numbers to every distinct word. By using this indexed corpus each token was reduced to their stems and again these stems were indexed by giving an index number to every stem and calculated their frequencies according to their occurrences in documents.

For finding stems of words we used Zemberek [22] which is a Turkish NLP library. Zemberek intends to provide library and applications for solving Turkish Natural Language Processing related computational problems. Turkish, by nature has a very different morphological and grammatical structure than Indo-European languages such as English. Since it is an agglutinative language like Finnish even making a simple spell checker is very challenging. Our corpus was resolved by Zemberek and obtained all the possible stems and grammatical types for each given word. Not all the time we could obtain the exact stem and grammatical type for the resolved words so we needed a module for reducing these possibilities to exact one. Zemberek would give a maximum of eight stems or grammatical types if the exact one could not found. So we assumed that each word might have maximum eight grammatical types. The system automatically detects the type by implementing the grammatical rules on the sentence. For this process, the following rules are implemented in the given order;

1. If a word's possible type counted more than one and all of them are same than this type is the words type.

2. If a word's possible types include an adjective, a word has no affix and the next word is a noun or a pronoun, then this word is an adjective.

3. If a word's possible types include an adjective and a word has an affix, the adjective is removed from that word's possible types and the number of types is decreased. If the number of types is fall down to one, then this type is considered to be that word's type.

4. If a word's possible types include an adverb rather than an adjective and the next word is a verb or the word is at the end of the sentence, then this word is an adverb.

5. If a word's possible types don't include adjective but adverb and the next word

is noun, adverb is removed from word's possible types and the number of type is decreased. If number of type is fall down to one this type is word's type.

6. If a word's possible types include not an adjective but an adverb and the word is at the end of the sentence, then this word is a verb.

7. If a word's possible type counted more than one and the type could not found then the type is the maximum counted one.

To detect the authorship features, a training set was formed from 20 different articles of 20 authors. 35 style markers have been figured out from all of these articles. By taking the average of each author we have collected a feature vector for each of the 20 authors. These vectors were converted into ARFF files.

An ARFF (Attribute-Relation File Format) file is an ASCII text file that describes a list of instances sharing a set of attributes. ARFF files were developed for use with the Weka [23] machine learning software. This document describes the version of ARFF used with Weka. ARFF files have two distinct sections. The first section is the header information, which is followed the data information. The header of the ARFF file contains the name of the relation, a list of the attributes (the columns in the data), and their types. Attribute declarations take the form of an ordered sequence of attribute statements. Each attribute in the data set has its own attribute statement which uniquely defines the name of that attribute and its data type. The order the attributes are declared indicates the column position in the data section of the file.

For measuring classification performance we used the technique known as k-fold cross validation to provide a more meaningful result by using all of the data in the data set as both training and test data. In this technique, the data is split into k folds, which are as equal in size as possible. A set of classifiers is then learnt from k-1 folds of data, with the remaining fold being used as the test set. This procedure is then repeated so that each fold is held out for testing. The results of the classifications from the k tests are combined to calculate the overall results for the data set. Most commonly, k is set equal to 10.

Results and Discussions

In order to test the accuracy of our module, we performed several experiments by using different parts of the corpus as training set and test set in each. In addition to calculating the performance of the proposed module, we also calculated its performance when only the morphological analyzer is used (without any statistical data from the corpus). We consider this a baseline performance and they are shown in Table 2.

Machine Learning Algorithm	Accuracy
Bayes Net	50 %
Naive Bayes	60.75 %
Naive Bayes Multinomial	65.75 %
Naive Bayes Updateable	60.75 %
Logistic	57.5 %
Multilayer Perceptron	63 %
RBF Network	66 %
Simple Logistic	69.25 %
SMO	58.5 %
Classification Via Regression	58.5 %
Decorate	59.75 %
Logit Boost	62.25 %
Multi Class Classifier	65.25 %
LMT	70.75 %
User Classifier	64 %

Table 2 Baseline Performances

We see that using statistical data greatly improves the performance of the module when compared with the baseline. We also observe that the success rates are not as high as expected. There are two reasons for this, the first one being the stems. Turkish has a very complex derivational and inflectional morphology. There are about 200 suffixes that can be attached to words and it is possible to derive several millions of words from a single root word. A word may change its part of speech freely by affixing different suffixes. The second and more important reason originates from the corpus size. We used a corpus with about 170,000 words, which is a very small size for our task. The difficulty involved in this task and in other statistical natural language processing tasks is acquiring a large enough and manually tagged corpus. Such corpora exist for widely used languages like English, but they are not available for Turkish.

An interesting problem is ”odd man out” in which check if a document belongs to any of the given set of authors, each of them being represented by a set of documents they authored. Given a list of authors, the “odd man out” task is to determine whether a particular document was written by one of these authors, or by someone else. Let us assume that there is a training set of documents available, where each document was written by one of the target authors, and that there is at

least one document written by each of those authors. It also seems natural to assume there are other documents available that do not belong to any of the target authors. We are going to use the authors of these other documents as "decoys" for training our classifier. Of course, it's better if these documents have much in common with available documents from the target authors. For the purpose of experimental work, all the documents will be taken from the same corpus.

To maximize the success of the author identification system we tested several different methods. We experimentally studied different sets of stylometric features and their combinations and found their relative value for classification to be fairly stable on diverse data collections.

First of all we performed our module on all of 35 style markers, however we did not obtain satisfactory results while all the style markers were used. Some of them had more deterministic values on author identification. So we eliminated some of them according to attribute evaluator functions of Weka. These functions were used with specific search algorithms. These functions and their search methods are shown in Table 3.

Attribute Evaluator	**Search Method**
CFS Subset Evaluator	Best First Search
CFS Subset Evaluator	Genetic Search
CFS Subset Evaluator	Greedy Stepwise Search
CFS Subset Evaluator	Rank Search
Consistency Subset Evaluator	Best First Search
Consistency Subset Evaluator	Genetic Search
Consistency Subset Evaluator	Greedy Stepwise Search
Consistency Subset Evaluator	Rank Search
Principal Components Evaluator	Ranker Search

Table 3 Attribute Evaluation

The most successful one was the CFS Subset Evaluator by using Rank Search method. With this approach we decremented the number of style markers from 35 to 22. Selected style markers were SM1-9, SM11, SM13, SM14, SM22-25, SM27, SM28, SM30, SM33-35. Remaining style markers were eliminated and they were not used in classification methods. Because the remaining style markers had a closer values on identification and resulting with wrong classifications. Results which were obtained by this method are shown in Table 4.

Machine Learning Algorithm	**Accuracy**
Bayes Net	53 %
Naive Bayes	69.25 %
Naive Bayes Multinomial	80 %
Naive Bayes Updateable	63.25 %
Logistic	62 %
Multilayer Perceptron	69.25 %
RBF Network	70.5 %
Simple Logistic	73 %
SMO	56.75 %
Classification Via Regression	59.25 %
Decorate	58.5 %
Logit Boost	59.5 %
Multi Class Classifier	57 %
LMT	72.5 %
User Classifier	69.5 %

Table 4 Results With Selected Attributes

Classifiers that we used to identify authors are shown in Table 5. There are four Bayes classification algorithms implemented in WEKA package as shown in Table 5. Neural networks can, depending on the problem domain, be limited to the use of a small number of features. The "Naive Bayes Multinomial" techniques may be suitable for use in authorship attribution. We obtained the highest success with this method.

Classifier	**Machine Learning Algorithm**
Bayes	Bayes Net
Bayes	Naive Bayes
Bayes	Naive Bayes Multinomial
Bayes	Naive Bayes Updateable
Functions	Logistic
Functions	Multilayer Perceptron
Functions	RBF Network
Functions	Simple Logistic

Functions	SMO
Meta	Classification Via Regression
Meta	Decorate
Meta	Logit Boost
Meta	Multi Class Classifier
Trees	LMT
Trees	User Classifier

Table 5 Classifiers

A Naive Bayes Multinomial classifier is a simple probabilistic classifier based on applying Bayes theorem with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be independent feature model. Depending on the precise nature of the probability model, Naive Bayes Multinomial classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for Naive Bayes Multinomial models uses the method of maximum likelihood; in other words, one can work with the Naive Bayes Multinomial model without believing in Bayesian probability or using any Bayesian methods.

Conclusions

As a conclusion a new classification technique is developed by the help of the known methods and it is compared with the known techniques. Our corpus had a training and a test sets each having 20 different authors. For identifying the authors, at the beginning 35 of style markers has been figure out. With this approach we have obtained a success rate of 70.75%. By using our method we selected 22 of style markers which were the most deterministic ones. We obtained the maximum success with Naive Bayes Multinomial which was 80% after attributes were eliminated using CFS Subset Evaluator with Rank Search method.

References

[1] A. Genkin, D. D. Lewis, and D. Madigan, Large-scale bayesian logistic regression for text categorization, 2004.

[2] B.Diri, M. F. Amasyalı, Automatic Author Detection for Turkish Text, ICANN/ICONIP'03 13th International Conference on Artificial Neural Network and 10th International Conference on Neural Information Processing, 2003.

[3] B.Kessler, G. Nunberg, H.Schutze, Automatic Detection of Text Genre, Proc. of 35th Annual Meeting of the Association for Computational Linguistics (ACL/EACL'97), 32-38 1997.

[4] Chris Callison-Burch, Co-training for Statistical Machine Translation, Master's thesis, University of Edinburgh, 2002.

[5] Christopher D. Manning and Hinrich Schütze, Foundations of Statistical Natural Language Processing, The MIT Press, 1999.

[6] D. Biber, Variations Across Speech and Writing, Cambridge University Press, 1988.

[7] D. I. Holmes, Stylometry: Its Origins, Development and Aspirations, presented to the Joint International Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing, Queen's University, Kingston, Ontario, 1997.

[8] D. Khmelev, Disputed authorship resolution using relative entropy for markov chain of letters in a text, In R. Baayen, editor, 4th Conference Int. Quantitative Linguistics Association, Prague, 2000.

[9] E. Stamatatos, N. Fakotakis, G. Kokkinakis, Automatic Text Categorization in Terms of Genre and Author, Computational Linguistics, pages 471-495, 2000.

[10] F. J. Tweedie, S. Singh, D. I. Holmes, Neural Network Applications in Stylometry: The Federalist Paper, Computers and the Humanities, Vol. 30, pages 1-10, 1996.

[11] H. Baayen, H. van Halteren, and F. Tweedie, Outside the cave of shadows: Using syntactic annotation to enhance authorship attribution. Literary and Linguistic Computing, Vol. 11(3), pages 121-131, 1996.

[12] J. Allen, Natural Language Understanding, Benjamin/Cummings Pub. Co., Redwood City, California, 1995.

[13] J. Burrows, Computation into Criticism: A Study of Jane Austen's Novels and an Experiment in Method, Clarendon Press, Oxford, 1987.

[14] J. Goldsmith, Unsupervised learning of the morphology of a natural language, Computational Linguistics, Vol. 27(2), pages 153–198, 2001.

[15] J. Karlgren, and D. Cutting, Recognizing Text Genres with Simple Metrics using Discriminant Analysis, Proceedings of the 15th. International Conference on Computational Linguistics, Kyoto, 1994.

[16] Jill M. Farringdon, Analyzing for Authorship: A Guide to the Cusum Technique. University of Wales Press, 1996.

[17] J. Rissanen, Stochastic Complexity in Statistical Inquiry, Volume 15. World Scientific Series in Computer Science, Singapore, 1989.

[18] Mathias Creutz, Unsupervised segmentation of words using prior distributions of morph length and frequency, In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL), pages 280–287, Sapporo, Japan, 2003.

[19] R. A. Bosch, J. A. Smith, Separating Hyperplanes and the Authorship of the Disputed Federalist Papers, American Mathematical Monthly, Volume 105, pages 601-608, 1998.

[20] S. Argamon-Engelson, M. Koppel, and G. Avneri, Style-based text categorization: What newspaper am I reading?, In Proc. AAAI Workshop on Learning for Text Categorization, pages 1-4, 1998.

[21] T. Mendenhall, The characteristic curves of composition, Science, 214:237249, 1887.

[22] https://zemberek.dev.java.net/

[23] http://www.cs.waikato.ac.nz/~ml/weka/index.html