A Variant of n-gram Based Language Classification

Andrija Tomović[1] and Predrag Janičić[2]

[1] Friedrich Miescher Institute for Biomedical Research
Part of the Novartis Research Foundation
Maulbeerstrasse 66, CH-4058 Basel, Switzerland
andrija.tomovic@fmi.ch
[2] Faculty of Mathematics, University of Belgrade,
Studentski trg 16,11000 Belgrade, Serbia
janicic@matf.bg.ac.yu

Abstract. Rapid classification of documents is of high-importance in many multilingual settings (such as international institutions or Internet search engines). This has been, for years, a well-known problem, addressed by different techniques, with excellent results. We address this problem by a simple n-grams based technique, a variation of techniques of this family. Our n-grams-based classification is very robust and successful, even for 20-fold classification, and even for short text strings. We give a detailed study for different lengths of strings and size of n-grams and we explore what classification parameters give the best performance. There is no requirement for vocabularies, but only for a few training documents. As a main corpus, we used a EU set of documents in 20 languages. Experimental comparison shows that our approach gives better results than four other popular approaches.

1 Introduction

The problems of similarities and dissimilarities between different languages and classification of multi-language documents have many everyday applications. One of them is automated classification of web pages, required, for instance, for restricting search only to documents written in a given language. Multilingual institutions, like EU, handle documents in more than twenty languages and rapid processing of such data is absolutely vital[3]. Automated classification of multi-language text has been studied for years and a variety of techniques give very good results.

In this paper we present a new variant of classification of multi-language documents based on n-grams. N-grams have been successfully used for a long time in a wide variety of problems and domains, including text compression, spelling

[3] There are 3400 people employed on translation and publications tasks in the European Commission. The annual budget for translations (between 23 languages) tasks is one billion Euros.

error detection and correction, information retrieval, automatic text categorization, authorship attribution, finding topical similarity between documents, but also in domains not related to language processing such as music representation, computational immunology, analysis of whole-genome protein sequences, protein classification and phylogenetic tree reconstruction (for more details and references see [14]).There are n-gram based techniques for distinguishing between documents written in different languages related to the work presented in this paper [3, 11]. Experimental comparison shows that our variation of n-grams-based classification is better than four other successful approaches to the language classification.

Overview of the paper. In Section 2 we give some basic definition and preliminary information. In Section 3 we describe our methodology for classification of multi-language documents, and the data we used for our analyses. In Section 4 we present and discuss our experimental results. In Section 5 we briefly discuss related work, and in Section 6 we present experimental comparison between our system and four other language classification tools. In Section 7 we draw final conclusions.

2 Preliminaries

In this section we give a brief overview of the notion of n-grams, of classification, and some algorithms for addressing this problem.

N-grams

Given a sequence of tokens $S = (s_1, s_2, ..., s_{N+(n-1)})$ over the token alphabet $\mathcal{A}$, where N and n are positive integers, an *n-gram* of the sequence S is any n-long subsequence of consecutive tokens. The i^{th} n-gram of S is the sequence $(s_i, s_{i+1}, ..., s_{i+n-1})$ [13]. Note that there are N such n-grams in S. There are $|\mathcal{A}|^n$ different n-grams over the alphabet $\mathcal{A}$ (where $|\mathcal{A}|$ is the size of $\mathcal{A}$).

For example, if $\mathcal{A}$ is the English alphabet, and l a string over the alphabet $\mathcal{A}$, $l =$ `life_is_a_miracle`, then 1-grams are: `l`, `i`, `f`, `e`, `_`, `i`, `s`, `a`, `m`, `r`, `c`; 2-grams are: `li`, `if`, `fe`, `e_`, `_i`, `is`, `s_`, `_a`, ...; 3-grams are: `lif`, `ife`, `fe_`, `e_i`, ... ; 4-grams are: `life`, `ife_`, `fe_i`, ... and so on.

When used in processing natural-language documents, n-grams show some of its good features:

- robustness: relatively insensitive to spelling variations/errors;
- completeness: token alphabet known in advance;
- domain independence: language and topic independent;
- efficiency: one pass processing;
- simplicity: no linguistic knowledge is required.

The problem with using n-grams is exponential combinatorial explosion. If A is the Latin alphabet with the space delimiter, then $|A| = 27$. If one distinguishes between upper and lower case letters, and also uses numerical digits, then $|A| = 63$. It is clear that many of algorithms with n-grams are computationally too expensive even for $n = 5$ or $n = 6$ (for instance, $63^5 \approx 10^9$).

Dissimilarity measures

Dissimilarity measure d is a function on two sets of texts $\mathcal{P}_1$ and $\mathcal{P}_2$ (defining specific *profiles*) and it should reflect the dissimilarity between these two. In the following text, by *(dis)similarity of texts* we denote a measure of (dis)similarity of two n-gram distributions.

In [2], some pioneer methods for the authorship attribution problem[4] and dissimilarity measures were discussed. For a range of language processing problems there were proposed techniques based on n-grams. For the authorship attribution problem, the bigram letter statistic was used: two texts are compared for the same authorship, using the dissimilarity formula:

$$d(M,N) = \sum_{I,J} [M(I,J) - E(I,J)] \cdot [N(I,J) - E(I,J)] \tag{1}$$

where I and J are indices over the range $\{1, 2, \ldots, 26\}$, i.e., all letters of the English alphabet; M and N are two texts written in the English alphabet; $M(I,J)$ and $N(I,J)$ are normalized character bigram frequencies for these texts and $E(I,J)$ is the same normalized frequency for "standard English". The technique is based on the following idea: the smaller $d(M,N)$, the more likely is that the author of the text N is the same as the author of the text M. As the bigram frequencies of "standard English" are obviously language-dependent parameters, another dissimilarity measure is given:

$$d(M,N) = \sum_{I,J} [M(I,J) - N(I,J)]^2 \ . \tag{2}$$

Following the ideas from [2, 8], a wide range of new dissimilarity functions were introduced and tested in [14]. The following functions performed best on different sets of problems:

$$d'(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} \frac{|f_1(n) - f_2(n)|}{\sqrt{f_1(n) \cdot f_2(n)} + 1} \tag{3}$$

$$d''(\mathcal{P}_1, \mathcal{P}_2) = \sqrt{\sum_{n \in profile} (f_1(n) - f_2(n))^2} \tag{4}$$

where *profile* is a set of all n-grams appearing in $\mathcal{P}_1$ or $\mathcal{P}_2$ and $f_i(n)$ is a normalized frequency for n-gram n in the set $\mathcal{P}_i$. While the function d'' is widely used, as far as we know, the dissimilarity function d'' was introduced recently in [14].

Classification

Given a set of objects, which is partitioned into a finite set of classes, *classification* is the task of automatically determining the class of an unseen object,

[4] The authorship attribution problem is as follows: given texts written by authors A_1, A_2, ... A_n, and one additional piece of text, guess who of the given authors wrote that piece of text.

based typically on a model trained on a set of objects with known class memberships. Classification is a *supervised* process, in a sense that it typically requires labelled training data to train a classifier.

We use the following simple classification method based on n-grams [14]: for a given set of families $\mathcal{P}_i$, $i = 1, 2, \ldots, k$ and the given object e, compute the dissimilarity measures $d(\{e\}, \mathcal{P}_i)$, $i = 1, 2, \ldots, k$. If the value $d(\{e\}, \mathcal{P}_s)$ is the smallest one, then the guess is that e belongs to the family $\mathcal{P}_s$. Thus, the classification algorithm is simple and its quality completely relies on the appropriateness of the dissimilarity measure used. This is essentially the well-known k Nearest Neighbours (kNN) classification method, with $k = 1$ [6].

3 Methodology and Data

For classification, we use the algorithm described in Section 2. For dissimilarity measure, we use the functions d' and d'' (as given by the equations (3) and (4)).

Our corpus is made out of documents in 20 European languages available from the EU web site[5]. For each language we took 20 documents for the corpus[6]. These are not necessarily translations of the same texts. Table 1 shows the list of these 20 languages and their codes.[7]

Table 1. Language codes

code	language	code	language
cs	Czech	lt	Lithuanian
da	Danish	hu	Hungarian
de	German	mt	Maltese
et	Estonian	nl	Dutch
el	Greek	pl	Polish
en	English	pt	Portuguese
es	Spanish	sk	Slovakian
fr	French	sl	Slovenian
it	Italian	fi	Finish
lv	Latvian	sv	Swedish

We consider the classification in the following way: for each language we randomly take 15 (out of 20) documents as a training corpus, for building n-gram language profiles. Then we classify the remaining 20×5 documents[8] and

[5] `http://europa.eu/`

[6] The whole corpus is available from:
`http://www.fmi.ch/members/andrija.tomovic/corpus-all.zip`.

[7] Complete Unicode table of language codes can be found at:
`http://unicode.org/onlinedat/languages.html`

[8] Test documents are available from:
`http://www.fmi.ch/members/andrija.tomovic/test.zip`.

count a percentage of correct guesses. The variant of this classification task is as follows: from the test documents (20×5 of them) we produce all subsequences of length L (L=$10, 20, 30, \ldots$) and apply the classification algorithm to these sequences. The motivation for this experiment is exploring the lower limits in length of documents for reliable classification.

The algorithm is implemented using Visual C# (Visual Studio 2003) within a wider application[9]. The application offers a range of functionalities concerning classification and clustering algorithms and the user can choose between a number of dissimilarity functions [14].

4 Experimental Results

The basic classification task (classification of 20×5 documents) is performed in the way described in Section 3. Table 2 shows success rates for dissimilarity functions d' and d''.[10] It can be seen that the success rate for both functions is perfect 100% for small values of n. The quality of the results is very high, especially taking into account that the classification is 20-fold (i.e., each test document was supposed to be classified into one of 20 categories). As expected, the success rate decreases as n increases (after some value). Indeed, long (e.g., 10 characters long) n-grams do not have high frequencies, so the classification results cannot be very stable. Despite that, success rate for d' remain 99% for $n = 8, 9, 10$, with only one error (one document in the Slovakian language classified as being in the Czech language). As observed for different domains in [14], the function d' performed better than d'', proving its quality in classification problems.

Table 2. Success rates for dissimilarity functions d' and d'' in classification of multi-language documents

n	1	2	3	4	5	6	7	8	9	10
d'	100%	100%	100%	100%	100%	100%	100%	99%	99%	99%
d''	100%	100%	100%	100%	100%	99%	92%	88%	84%	28%

The above results show excellent success rate for classification of whole documents. The question is whether shorter texts would also be successfully classified. Of course, very short strings (e.g., up to 20 characters) are very unlikely to be classified with high success rate (since n-gram frequencies in one short string can be very different than frequencies in the whole of the language). It is interesting

[9] The application is available from:
`http://www.fmi.ch/members/andrija.tomovic/NgramsApplication.zip`

[10] All training data and the detailed experimental results can be obtained upon request from the first author.

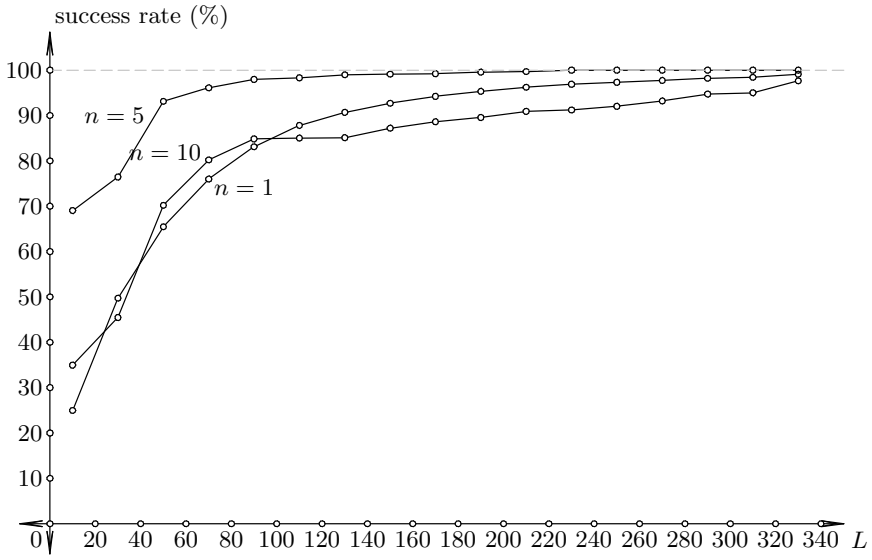

Fig. 1. Success rate for classification of strings of length L, by using n-grams with different values of n.

to explore what string length is sufficient for obtaining high (say, more than 99%) success rate in classification. The following experiment is aimed at answering this question. As in the previous experiment, for each language we take 15 (out of 20) documents as a training corpus, for building the n-gram language profile. Then we construct *all* L-substrings (L-grams) from the remaining 20×5 documents and count a percentage of correct guesses for these L-grams, as for strings to be classified. For classification of these strings, we used n-grams for different values of n. Figure 1 shows the results for n=$1, 5, 10$, for L=$10, 30, \ldots$ For all values of n, almost perfect success rate is reached for L as small as 300. For all values of L the success rate was best for n= 5. For n=5, 93% success rate is reached for L as small as 50, and 99% success rate is reached for L equal 150. The value n=5 gives the best results because:

- Short n-grams cannot distinguish different languages easily. Namely, different languages with similar alphabets can have similar frequencies for some n-grams, and hence, a short test text, with non-representative distribution of n-grams (non-representative w.r.t. the language it belongs to), can be wrongly classified.
- Long n-grams have lower and lower frequencies and become more and more language- and string-specific. Table 3 and Table 4 show the first 10 most frequent n-grams in training data set for Italian and English. It can be seen that with higher n, n-grams become highly dependent on the training set. Long n-grams make better distinguishing between different languages, but

on the other hand, text from the same language which is not from the same domain as the training data set is difficult to be recognized (this is well-known *over-fitting problem*).

Table 3. The most frequent n-grams (top 10) in training data for the Italian language

n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10
_	e_	_de	_di_	ione_	zione_	azione_	zione_de	ommission	ommissione
i	i_	_di	ione	azion	azione	zione_d	azione_d	mmissione	_Commissio
e	a_	ion	zion	zione	ione_d	ione_de	ione_del	_Commissi	Commission
a	_d	re_	one_	ation	e_dell	_della_	missione	Commissio	a_Commissi
o	o_	di_	_del	_dell	mento_	_europe	mmission	zione_del	mmissione_
t	on	one	azio	_che_	one_de	amento_	ommissio	a_Commiss	la_Commiss
r	re	ne_	che_	e_del	amento	_delle_	Commissi	missione_	zione_dell
n	ti	to_	dell	e_di_	sione_	one_del	_Commiss	ione_dell	azione_del
l	_c	zio	_con	_del_	_della	ssione_	_deputat	azione_de	terrorismo
s	er	_co	_la_	_del_	della_	issione	a_Commis	_terroris	_terrorism

Table 4. The most frequent n-grams (top 10) in training data for the English language

n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10
_	e_	_th	_the	_the_	n_the_	_of_the	_of_the_	European_	_European_
e	_t	the	_of_	_and_	_of_th	of_the_	European	_European	Parliament
t	th	he_	_to_	tion_	of_the	_Europe	uropean_	arliament	Commission
o	_a	ion	and_	ation	_that_	uropean	_on_the_	Parliamen	arliament_
a	he	on_	_and	n_the	f_the_	Europea	_Europea	_terroris	_Commissio
i	s_	_to	tion	_of_t	ation_	_on_the	rliament	ommission	e_European
n	n_	to_	ing_	_for_	Europe	ropean_	arliamen	Commissio	e_Commissi
r	d_	_of	ion_	of_th	_Europ	on_the_	_in_the_	rliament_	he_Commiss
s	in	of_	ment	f_the	_the_E	in_the_	Parliame	_Commissi	_Parliamen
h	on	_an	_in_	that_	uropea	_in_the	terroris	e_Europea	he_Europea

The results shown in Figure 1 also suggest a general classification strategy: if the string that is to be classified is shorter than 350, apply the classification procedure with n=5, otherwise apply the classification procedure with n=1 (because the success rates are almost equal for n=5 and n=1, and the classification for larger n is more time-consuming).

In order to make a system more efficient, the option is to take into account only a certain number of most frequent n-grams (instead of considering all occurring n-grams). However, in that case, classification with the functions d' and

d'' is not perfect for all values of n. The results of classification performed using the first 100 most frequent n-grams are given in Table 5. For 1-grams results are the same (perfect 100%) like in Table 2 because there are no more than 100 1-grams. For 2-grams, 3-grams, 4-grams results are good, because first 100 most frequent n-grams (n=2,3,4) represents a significant portion of the set of all 2,3,4-grams. In these cases, the result of the training phase is a set of only $20 \times 100 \times n$ characters, yielding a knowledge sufficient for perfect identification of 20 languages.

Table 5. Success rates for dissimilarity functions d' and d'' in classification of multi-language documents using only the first 100 most frequent n-grams

n	1	2	3	4	5	6	7	8	9	10
d'	100%	100%	100%	100%	90%	68%	54%	35%	35%	27%
d''	100%	100%	100%	100%	94%	76%	69%	59%	54%	51%

We can conclude that our n-grams based technique gives excellent results, even for 20-fold classification of documents in different languages, and even for short strings. For this sort of classification, there is no need for vocabularies, but only for a very small amount of training data. For training data we used only 15, rather short documents for each of 20 languages. For instance, training data for English had a total size of only 52Kb and around 2000 different words (including different forms).

It is interesting to report on the dissimilarities (based on the given functions) between profiles for different languages. To a somewhat surprise, these dissimilarities are not in accordance with the traditional clustering of languages — for instance, the English language is closer to the Italic languages than to the Germanic languages (this is true for both dissimilarity functions, for all values of n, and for the variant with 100 most frequent n-grams used). A possible explanation for this could be that written languages and distributions of their n-grams do not reflect deeper relationships between spoken languages.

5 Related Work

Automated classification has been studied for years and there is a number of methods for these problem. Also, there are many techniques for classification of documents in different languages and many of them give excellent results. However, some of them use some language-specific knowledge, some are applied to specific corpora, some are applied over specific sets of languages, and so it is not easy to make a direct, relevant comparison between different approaches. In [12] there is a good overview of different approaches to this problem, including approaches based on the presence of specific characters, on the presence of

specific letter combinations, on the presence of specific words, on distribution of n-grams, etc.

The technique presented in [3] is based on using an ad hoc rank order statistic to compare the prevalence of n-grams. The test and training texts are first tokenized in order to avoid sequences which straddle two words. Comparison between test and training profiles was based on comparing rankings of the most frequent n-grams. There are results for 8-fold classification (over 8 languages), and for test strings long 300 characters or more.

A n-gram-based Bayesian classifier is described in [7]. The technique is domain independent and does not require tokenization. There are results only for 2-fold classification (over English and Spanish). As expected, classification success rate is higher with longer training data and longer test data, and generally better results are obtained for bigrams and trigrams. For 50Kb of training sets and for 20 bytes of input string, the success rate was 92%, while for 500 bytes of input string, the success rate was even 99.9%.

Approach based on n-grams, described in [10], uses simple information theoretic principles (perplexity and entropy) in combination with Bayesian decision theory. That technique has been evaluated on four different languages and four different text categorization problems.

The technique presented in this paper is similar to the one from [12], but based on different dissimilarity functions. The technique from [12] is applied to 18-fold language classification and it performs well even with short training and test data, it is simple and easy to implement. The success rate goes from 78.2% for 1-grams, for 200 lines of training data, and 1 line of test data, up to 100% for 2-grams, 2000 lines of training data, and 20 lines of test data.

In a recent paper [9], the problem of identifying language in web documents is addressed, for which the authors claim that it is more difficult variant of the problem. The authors use n-grams and several dissimilarity measures and reach around 91% success rates for 12-fold language classification.

There is another recent paper [4], addressing the problem of language identification and some of its hard variants: including distinguishing between European and Brazilian variants of Portuguese and identifying small tourists advertisements. The proposed, n-gram based, approach was also used for the standard language identification problem: the system was trained by 235 documents written in 19 European languages, and tested by 290 new test samples — once of size at least 6 lines (with 100% success rate) and once of size 4 lines (with 98.9% success rate). The system was also tested for identifying European and Brazilian variants of Portuguese (with 200 training documents and 369 test documents), reaching 98.37% success rate.

6 Experimental Comparison to Other Tools

We performed the experimental comparison between our system and several well-known tools:

Xerox language identifier is based on [1]. The algorithm performs the classification by calculating probabilities, based on n-gram probabilities for each language from a training data set. It has support for 47 languages.

Unknown language identification [11] is based on the algorithm described in [5]. The method uses a vector with frequencies of all n-grams for training profiles. The distance function is defined as cosine function of the angle between the sample text vector and each of text vector from the library. It has support for 66 languages.

TextCat is an implementation of the text categorization method described in [3]. This algorithm is similar to the algorithm proposed here. A preprocessing of the input text is performed, and digits and punctation are discarded. In order to generate a profile, the method uses all n-grams for n=1,2,3,4,5 on preprocessed text. Then the first 300 most frequent n-grams are used by different methods for comparing n-gram profiles. It has support for 77 languages.

SILC [12] uses Bayesian decision theory and classic Noisy Channel statical approach. This approach is different from our method and uses only 1-grams and 3-grams. It has support for 39 languages.

Table 6. Experimental results of four language classification tools

Tool	success rate	comment
XEROX	99.00% (99/100)	
ULI	98.75% (79/80)	no support for el,mt,sl,sk
TextCat	97.89% (93/95)	no support for mt
SILC	97.33% (73/75)	no support for lt,lv,mt,sk,sl

We used the corpus and the test data described in Section 3.[13] For testing the above tools, we used the set of 100 documents that we used as a test set for our system, so — all systems were ran on the same documents. Some of the above four tools do not have support for some languages, so the total of available tests was lower for some tools. Table 6 shows the experimental results. As it can be

[11] `http://complingone.georgetown.edu/~langid/`

[12] `http://rali.iro.umontreal.ca/`

[13] Unfortunately, for most of approaches described in Section 5, data sets and software which authors used are not publicly, freely available. This makes it difficult to perform a fair comparison between different existing tools and further evaluation of the presented method. Trying to promote another practice, we provide all data and tools which we used publicly availably. A good practice is also using rich sources of freely available multilingual corpora (and specifying used subsets), such as the one we use in this paper (`http://europa.eu/`, `http://eur-lex.europa.eu/`), or collections of translations of the Bible (`http://bibledatabase.net/`, `http://www.biblegateway.com/`).

seen, all the tools performed in an excellent way, however none of them reached perfect success, unlike our approach (see the results in Table 2 and Table 5). Despite the fact that the above tools have support for more languages than our system, the given results are still relevant and fair. Namely, almost all wrong (5 out of 6) classifications made by the tested tools pointed to the languages that our system also has support for(with one exception for SCI). The most frequent error (4 out of 6) was wrongly classifying czech documents as slovakian and vice versa.

For further evaluation of our method we also used one chapter of the Bible like in work [10]. In that work authors used translation of one chapter into 6 different languages and achieved 100% accuracy. We used one chapter (Ruth) in 10 different languages. We have used fist two subchapter for the training data set and the rest for test data. Our approach has achieved 100% accuracy for all size n-grams.

7 Conclusions

We presented a new variant of a language classification algorithm based on n-grams (using the dissimilarity function recently introduced in [14]). We analyzed its performance on the test documents written in 20 languages. The results are very good, and they reach perfect 100% success rate for our corpus of integral documents (for small size of n-grams). There is also high success rate for very short fragments of test text, reaching 99%, while longer fragments were classified with 100% success rate. These are very good results, especially taking into account that the classification is 20-fold. This sort of classification does not require massive vocabularies, but only very few training documents in different languages. We made an experimental comparison of our tool with four other language classification tools, and it gave the best performance. Our classification mechanism is fast, robust, and does not require any knowledge of the different languages. We believe it can be used in different contexts, like in multi-lingual institutions, and in Internet search engines.

References

[1] K. R. Beesley. Language Identifier: A Computer Program for Automatic Natural-Language Identification of On-Line Text. In *Languages at Crossroads: Proceeding of the 29 th Annual Conference of the American Translator Association*, pages 47–54, 1988.

[2] W. R. Bennett. *Scientific and engineering problem-solving with the computer*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1976.

[3] W. B. Cavnar and J. M. Trenkle. N-gram-based text categorization. In *Proceedings of the 1994 Symposium On Document Analysis and Information Retrieval*, University of Nevada, Las Vegas, April 1994.

[4] J. F. da Silva and G. P. Lopes. Identification of document language is not yet a completely solved problem. In *Internationa Conference on Computational Intelligence for Modeling, Control & Automation (CIMCA)*. Computer Society, IEEE, 2006.

[5] M. Damashek. Gauging similarity with n-grams: Language independent categorization of text. *Science*, 267:843–848, 1995.

[6] M. H. Dunham. *Data Mining Introduction and Advanced Topics*. Southern Methodist University, Pearson Education Inc., New Jersey, 2003.

[7] T. Dunning. Statistical Identification of Language. Technical Report Technical report. CRL MCCS-94-273, Computing Research Lab, New Mexico State University, 1994.

[8] V. Kešelj, F. Peng, N. Cercone, and C. Thomas. N-gram-based author profiles for authorship attribution. In *Proceedings of the Conference Pacific Association for Computational Linguistics, PACLING'03*, Dalhousie University, Halifax, Nova Scotia, Canada, August 2003.

[9] B. Martins and M. J. Silva. Language identification in web pages. In *Proceedings of the 2005 ACM symposium on Applied computing*. ACM Press, 2006.

[10] F. Pegn, D. Schuurmans, and S. Wang. Language and Task Independent Text Categorization with Simple Language Models. In *Proceedings of Human Language Technology Conference*, pages 110–117, 2003.

[11] J. Schmitt. Trigram-based method of language identification. In *U.S. Patent number:5062143*, October 1991.

[12] P. Sibun and J. C. Reynar. Language identification: Examining the issues. In *Proceedings of 5th Symposium on Document Analysis and Information Retrieval*, 1996.

[13] D. Tauritz. Application of n-grams. Department of Computer Science, University of Missouri-Rolla.

[14] A. Tomović, P. Janičić, and V. Kešelj. N-gram-based classification and unsupervised hierarchical clustering of genome sequences. *Computer Methods and Programs in Biomedicine*, 81(2):137–153, 2006.