Language Independent n-Gram-Based Text Categorization with Weighting Factors: A Case Study

Jelena Graovac[1], Jovana Kovačević[1,2], Gordana Pavlović-Lažetić[1]

[1] University of Belgrade, Faculty of Mathematics, Department of Computer Science,
Studentski trg 16, 11000 Belgrade, Serbia
{jgraovac,jovana,gordana}@matf.bg.ac.rs
[2] Indiana University, School of Informatics and Computing,
Bloomington, Indiana, USA

Abstract. We introduce a new language independent text categorization technique based on byte-level n-gram profiles, an n-gram weighting factors scheme, and a simple algorithm for comparing profiles. The technique does not require any morphological analysis of texts, any preprocessing steps, or any prior information about document content or language. We apply it to the text categorization problem in two widely spoken yet paradigmatically quite different languages – English and Arabic, thus demonstrating language-independence. We used their publicly available document collections – 20-Newsgroups and Mesleh-10, respectively. Experimental results presented in terms of macro- and micro-averaged $F1$ measures imply that the new technique outperforms other n-gram based and bag-of-words machine learning techniques when applied to English and Arabic text categorization.

Categories and Subject Descriptors: I.7 [**Document and Text Processing**]: Miscellaneous

Keywords: Arabic, byte-level n-gram, English, kNN, natural language text categorization

1. INTRODUCTION

Automated text categorization (TC) is a supervised learning task, defined as assigning (pre-defined) category labels to new documents based on the likelihood suggested by a set of labelled documents [Yang and Liu 1999]. Such a process has different useful applications including, but not restricted to, document organization, text filtering, spam detection, mail routing, news monitoring, automatic document indexing and a hierarchal catalogue of web resources [Duwairi 2007]. The rapid growth of the Internet has increased the number of online natural language text documents available. A portion of these documents are already classified into specific categories by the authors or publishers of the texts. However, the amount of yet unclassified is still too large. Since building text classifiers by hand is difficult and time-consuming, it is advantageous to build an automatic text classifier by learning from a set of previously classified documents [Sebastiani 2002]. After Lewis' influential thesis [Lewis 1992], the use of machine learning techniques for TC has gained in popularity. Some of them are k-Nearest Neighbors (kNN), Support Vector Machines (SVM), Decision Trees (DT), Rule-based classifiers, Naïve Bayes (NB), Rocchio's algorithm, Conceptual Structure, Neural networks, Genetic algorithms, Latent semantic analysis, Centroid based classifier, Conditional random fields, and Hidden Markov Models (HMM).

The problem of TC faces different challenges. One of them is presence of different kinds of textual errors, such as typing, spelling and grammar errors. TC has to perform reliably on all inputs, and thus has to tolerate these kinds of problems to some extent. Although most of the research activity has concentrated on English text, TC in other languages is also an important area of research. Using

different languages produces additional difficulty in TC procedures regarding specific features of the languages. TC in Arabic, for example, presents a particular challenge. Arabic is one of the most widely spoken languages in the world and it is the mother tongue of more than 300 million people [Kourdi et al. 2004]. Arabic belongs to the Semitic language family. It is written from right to left and consists of 28 letters. It is a highly inflectional and derivational language, which makes morphological analysis a very complex task. This complex morphology usually creates the necessity to apply a set of preprocessing activities to documents before they become suitable for manipulation. Moreover, some vowels in Arabic are represented by diacritics that are usually removed in the preprocessing phase which leads to great ambiguity. Also, Arabic scripts do not use capitalization for proper nouns [Al-Shalabi and Obeidat 2008]. These and many other challenges in analyzing the Arabic language are enumerated and detailed in [Dichy 2002]. Many powerful techniques show some disadvantages when dealing with languages other than English, such as using some language-specific knowledge or requiring some non-trivial text preprocessing steps [Kursat and Serkan 2014].

The goal of this article is twofold: first, to present a new n-gram based language independent TC technique that avoids many of the above mentioned difficulties; second, to test and compare its performance on TC in two of the most influential and paradigmatically quite different languages – English and Arabic, thus demonstrating its language-independence. Except English, we choose to work with Arabic as the language that is widely spoken and very different from English. Also, it is among top ten languages most used in the Internet according to the Internet World State rank[1]. One of the great advantages of the technique that is presented in this article is its fully topic and language independence, so it can be equally well applied to any other language, without any changes.

We now give a brief outline of the article. Some background information is presented in Section 2 and Section 3 gives a discussion of related work. Section 4 describes methodology for TC used in this article. This section also presents several dissimilarity measures, the datasets used for TC and the set of evaluation metrics that are used to assess the performance of this technique. Section 5 reports on experimental results and shows comparisons of dissimilarity measures. We compare our results with the results obtained by other n-gram based and bag-of-words state-of-the-art methods for English and Arabic. Finally, Section 6 concludes the article.

2. BACKGROUND

2.1 Text Categorization Problem

TC is the task of classifying unlabelled natural language documents into a predefined set of categories. Formally, TC consists in associating a Boolean value to each pair $(d_j, c_i) \in D \times C$ where D is the set of text documents and C is the set of categories. The value T (Truth) is then associated to the pair (d_j, c_i) if the text document d_j belongs to the category c_i while the value F (False) is associated to it otherwise. The goal of the TC is to approximate the unknown target function $\check{\Phi} : D \times C \to \{T, F\}$ (that describes how documents ought to be classified) by means of a function $\Phi : D \times C \to \{T, F\}$ called the classifier, such that $\check{\Phi}$ and Φ "coincide as much as possible" [Sebastiani 2002]. In the traditional machine-learning setting, each document $d_j \in D$ is assigned a single category $c_i \in C$. In the multi-label case, each document $d_j \in D$ may be assigned multiple labels in C.

2.2 Document Representation

The role of the document representation component is to represent text document so as to facilitate machine manipulation but also to retain as much information as needed. Text documents should be transferred into a compact and an applicable representation which is used uniformly in training, validation and classification. A text document d_j is usually represented as a vector of term weights

[1] http://www.internetworldstats.com/stats7.htm

$\overrightarrow{d_j} = (w_{1j}, w_{2j}, ...w_{|T|j})$ where T is the set of terms that occur at least once in at least one document of training set, and $0 \leq w_{kj} \leq 1$ represents, loosely speaking, how much term t_k contributes to the semantics of document d_j [Sebastiani 2002]. A common, and often overwhelming, characteristic of text data is its extremely high dimensionality. Feature extraction and feature selection techniques are widely employed to reduce the dimensionality of data and to enhance the discriminatory information. The word "feature" usually has two different but closely related meanings in the context of TC. One refers to which unit is used to represent or to index a document, while the other focuses on how to assign an appropriate weight to a given term. For the second meaning, the weight assigned to the given term comes from two sources: intra-document and inter-document. The intra-document based weight uses information within a document, while the inter-document based weight uses information in the corpus.

A typical choice for the first meaning of "feature" is to identify terms with words. This is often called either the set-of-words or the bag-of-words (BOW) approach to document representation, depending on whether weights are binary or not. In the case of non-binary indexing, for determining the weight w_{kj} of term t_k in document d_j any indexing technique that represents a document as a vector of weighted terms may be used, such as the standard *tf-idf* (term frequency-inverse document frequency) [Salton and Buckley 1988]. The *tf* part can be regarded as a weight from intra-document source, while the *idf* part is a weight from inter-document source. The major limitation of BOW technique is that the information about the sequence gets lost. Unlike this approach, n-gram techniques are based on n-grams that are n-contiguous sequences of bytes, characters or words.

2.3 N-grams

An n-gram is a sequence of consecutive symbols extracted from a long string. The symbol can be a byte, a character or a word. Extracting byte n-grams from a document is like moving an n-byte wide "window" across the document, byte by byte. Each window position covers n bytes, defining a single n-gram. In the case of Latin-alphabet languages, character-level and byte-level n-gram models are quite similar according to the fact that one character is usually represented by one byte. The only difference is that character-level n-grams use letters only and typically ignore digits, punctuation, and whitespace while byte-level n-grams use all printing and non-printing characters. For example, the string "Byte n-grams!" would be composed of the following byte-level 2-grams: By; yt; te; e_; _n; n-; -g; gr; ra; am; ms; s!. The underscore character ("_") is used here to represent space, and (";") character is used for separation of n-grams.

N-gram techniques have been successfully used for a long time in a wide variety of problems and domains. In natural language processing they turn out to be effective in many applications, including text compression, spelling error detection and correction, information retrieval, language identification, authorship attribution, topic-based TC etc. They also have proven to be efficient in domains not related to language processing such as protein categorization, computational immunology, music representation etc.

The use of byte n-grams has a lot of advantages:

—*Language and topic independence.* There is no need for any text preprocessing or higher level processing, such as tagging, parsing, or other language dependent and nontrivial natural language processing tasks.

—*Relative insensitivity to spelling variations/errors.* Since every string is decomposed into small parts, any errors that are present tend to affect only a limited number of those parts, leaving the remainder intact.

—*Word stemming is got essentially for free.* The n-grams for related forms of a word (e.g., "advance", "advanced", "advancing", "advancement", etc.) intrinsically have a lot in common when viewed as sets of n-grams.

—*No linguistic knowledge is required.* The information is not required even about space character used for word separation, the new line character, uppercase and lowercase letters, and the like.

—*Independence of alphabet.* In the case of byte-level n-grams, text is simply treated as a sequence of bytes instead of characters.

—*Only one pass processing is required.*

The main disadvantage of using n-gram technique is that it yields a large number of n-grams.

3. RELATED WORK

There has been immense work on TC in English using many different algorithms. It would be impossible for us to enumerate all the techniques that are used. Instead, we mention only previously published n-gram based methods and methods based on the bag-of-words (BOW) state-of-the-art (SOA) models, applied to the same corpus in English (the 20-Newsgroups) that we use in this article. Although most of the developed TC techniques are referred to English language, there are also a lot of research with significant results, performed on TC in Arabic.

N-gram methods for English (the 20-Newsgroups corpus). Character n-gram method was used for document representation in [Rahmoun and Elberrichi 2007] in order to solve the problem of TC in English. The effects of this method are examined in several experiments using the multivariate chi-square to reduce the dimensionality, the cosine and Kullback&Liebler dissimilarity measures, and two benchmark corpora – the 20-Newsgroups and the Reuters-21578, for evaluation. The results shown the effectiveness of this approach compared to the BOW and stem representations.

SOA methods for English (the 20-Newsgroups corpus). Significant results of the TC on English corpus 20-Newsgroups are achieved by Lan et al. [2009]. They represented text documents as BOW and they have investigated several widely-used unsupervised and supervised term weighting methods in combination with SVM and kNN algorithms. They introduced new technique called *tf-rf* (term frequency - relevance frequency) which have proved to be more effective than others.

N-gram methods for Arabic. The character-level n-gram technique was used in [Khreisat 2006] and [Sawaf et al. 2001] to classify Arabic newspapers. In [Sawaf et al. 2001] a statistical method called Maximum entropy was used. This statistical method was also used by El-Halees [2007]. In [Al-Shalabi and Obeidat 2008] the authors presented the results of classifying Arabic document set introduced by Mesleh [2007], using two kNN classifiers. The first classifier used word-level 1-grams and 2-grams while the second one represented a document as a BOW. Results shown that using n-grams to represent document produces better performance than using BOW model.

SOA methods for Arabic. The kNN algorithm and different variations of Vector space model and Term weighting approaches were investigated by Thabtah et al. [2008]. In [Syiam et al. 2006], the kNN and Rocchio algorithms were used, while kNN, Rocchio, and NB algorithms were used in [Kanaan et al. 2009]. The application of SVM to Arabic TC was presented in [Mesleh 2011] and [Mesleh 2007]. The results showed that the SVM algorithm with the chi-square method has outperformed NB and the kNN classifiers in terms of F1 measure. SVM was also used in [Gharib et al. 2009] while NB machine learning technique was used in [Kourdi et al. 2004]. Duwairi [2007] compared the performance of three classifiers for Arabic TC: kNN, NB and Distance-based classifiers. The NB classifier based on chi-square features selection method was used in [Thabtah et al. 2009]. Noaman et al. [2010] showed that using the root-based stemmer with NB classifier decreases the dimensionality of the training documents. An intelligent system based on statistical learning for searching in Arabic text was presented in [Althubaity et al. 2008]. The light stemmer was used for preprocessing, HMM for feature extraction, and NB classifier for categorization. In [Harrag et al. 2009] and [Saad and Ashour 2010] the DT algorithm was used. Saad and Ashour [2010] studied the impact of text preprocessing and different term weighting schemes on Arabic TC. In addition, they have developed

new combinations of term weighting schemes to be applied to Arabic text for TC purposes. In [Al-Harbi et al. 2008] two learning algorithms: the C5.0 DT and SVM with the BOW representation were compared. Raheel et al. [2009] used the technique of Boosting [Freund and Schapire 1996] in combination with DTs. In [Al-Diabat 2012] the authors tested different categorization data mining algorithms (C4.5, PART, RIPPER, OneRule) in order to solve the problem of Arabic TC. The results revealed that the least applicable learning algorithm to the chosen Arabic dataset is OneRule, and the most applicable algorithm is PART. Harrag et al. [2009] used Artificial Neural Network for the TC of Arabic text documents. In [Ismail et al. 2014] the authors compared five best known algorithms for Arabic TC. They also studied the effects of utilizing different Arabic stemmers (light and root-based stemmers) on the effectiveness of these classifiers.

4. METHODOLOGY AND DATA

The new n-gram TC technique that is presented in this article is an improved variant of the basic n-gram technique, presented and used by Keselj et al. [2003] to solve the authorship attribution problem. The technique is based on byte-level n-gram frequency statistics method for document representation, and kNN ($k = 1$) algorithm for the TC process. It is fully language and topic independent. In [Graovac 2014], this basic technique was used to solve the TC problem in Serbian, English and Chinese, while in [Graovac and Lažetić 2014] it is used to solve the problem of sentiment polarity detection in movie reviews in English and Spanish. As opposed to the basic technique where only normalized frequencies of n-grams are used for representing each category, the new technique employs an n-gram weighting factors schema, which makes it possible to represent the importance of n-grams in the concerned category, taking into account other categories as well. We are interested in n-grams with high frequency in the concerned category, but with low frequency in the whole corpus. While normalized frequencies come from intra-document source, weighting factors come from the inter-document source. In the rest of this article we use BnGT to denote "Basic n-Gram Technique" and WnGT to denote "Weighted n-Gram Technique" proposed in this work.

The steps of the TC procedure of the WnGT are as follows:

—For a given classified text corpus divided into training and test data, concatenate all the training documents that belong to the same category into a single document. Each category is thereby presented by one document only.

—For each category document and test document, construct its profile:

  —Select a specific n-gram size (e.g. 6-gram, 7-gram etc.).

  —Extract the byte-level n-grams for that particular value of n and calculate the normalized (relative) frequencies, for each n-gram.

  —List the n-grams by descending frequency, so that the most frequent are listed first. Category and document profiles have varying lengths depending on the length of the input data and the size of the n-gram.

  —Cut off all category profiles at the length of the shortest one. This will ensure that all category profiles will contain the same (maximum possible) number of n-grams.

  —For each n-gram x in each category profile compute its weighting factor:

$$w(x) = \frac{|C|^2}{c_f^2} \tag{1}$$

  where $|C|$ is the number of categories in corpus and c_f is the number of categories whose profiles contain the n-gram x.

  —Select a specific profile length L at which to cut off all document and category profiles.

—Assign each test document one or more categories:

—Compute a dissimilarity measure between the test document's profile and each of the category's profiles.
—Select the category whose profile has the smallest value of dissimilarity measure with the document's profile. If there are more than one such category, select them all.

Following this procedure, a test document profile is simply a set of L pairs $\{(x_1, f_1), (x_2, f_2),...,(x_L, f_L)\}$ of the most frequent n-grams and their normalized frequencies, while category profile is a set of L pairs $\{(x_1, f_1, w_1), (x_2, f_2, w_2), ..., (x_L, f_L, w_L)\}$ of the most frequent n-grams, their normalized frequencies (*tf* part) and their weighting factors (*idf* part), generated from training data. In order to decide whether a certain test document belongs (or not) to a certain category, this TC procedure requires a dissimilarity measure.

4.1 Dissimilarity measures

In this article, four dissimilarity measures are used. First of them is the modification of the measure presented by Keselj et al. [2003] that has a form of relative distance:

$$dmod(\mathcal{P}_1, \mathcal{P}_2) = \sum_{x \in profile} \Big(\frac{2 \cdot (f_1(x)w_1(x) - f_2(x))}{f_1(x)w_1(x) + f_2(x)} \Big)^2 \tag{2}$$

where $f_1(x)$ and $f_2(x)$ are frequencies of an n-gram x in the category profile $\mathcal{P}_1$ and the test document profile $\mathcal{P}_2$, respectively.

The next three measures are the modification of the measures that performed best on the topic-based TC problem, considering all 19 measures presented by Tomović et al. [2006]. First of them is the variation of the *dmod* measure where frequency differences are divided by the "average" (arithmetic mean value) frequency for a given n-gram:

$$d_1mod(\mathcal{P}_1, \mathcal{P}_2) = \sum_{x \in profile} \frac{2 \cdot |f_1(x)w_1(x) - f_2(x)|}{f_1(x)w_1(x) + f_2(x)} \tag{3}$$

The following two measures are based on the quadratic mean value:

$$d_2mod(\mathcal{P}_1, \mathcal{P}_2) = \sum_{x \in profile} \Big(\frac{\sqrt{2} \cdot |f_1(x)w_1(x) - f_2(x)|}{\sqrt{(f_1(x)w_1(x))^2 + f_2(x)^2}} \Big)^2 \tag{4}$$

$$d_3mod(\mathcal{P}_1, \mathcal{P}_2) = \sum_{x \in profile} \frac{\sqrt{2} \cdot |f_1(x)w_1(x) - f_2(x)|}{\sqrt{(f_1(x)w_1(x))^2 + f_2(x)^2}} \tag{5}$$

Modifications of these measures are reflected in the weighting factors $w_1(x)$ being added to n-grams in each category profile. In this way, n-gram that belongs to a smaller number of categories has a greater significance for the corresponding category.

Implementation Details. For producing n-grams and their normalized frequencies, the software package *Ngrams* written by Keselj et al. [2003] is used. For the process of TC, the software package *NgramsCategorization* developed by the authors of this article is used. Source code can be obtained on request from the first author.

4.2 Data Collections

For the empirical evaluation of the technique presented in this article, we used two benchmark text corpora in English and Arabic: the 20-Newsgroups and the Mesleh-10, respectively.

20-Newsgroups corpus in English. The 20 Newsgroups dataset is a collection of approximately 20000 newsgroup documents, evenly divided into 20 different newsgroups, each corresponding to a different topic. It was first collected by Lang [2004]. Three versions of this dataset are publicly available. The most popular is "bydate" version. It is sorted by date into training (60%) and testing (40%) sets. This is the corpus edition that is used for testing TC technique presented in this article. 20-Newsgroups corpus is a single label. Some of the newsgroups are very closely related to each other (e.g. comp.sys.ibm.pc.hardware/comp.sys.mac.hardware), while others are highly unrelated (e.g. misc.forsale/soc.religion.christian).

Mesleh-10 corpus in Arabic. This text corpus is introduced by Mesleh [2011] and it is divided into 10 categories, so we refer to it as Mesleh-10. It is collected from online Arabic newspaper archives, including Al-Jazeera, Al-Nahar, Al-hayat, Al-Ahram, Al-Dostor and a few other specialized web sites. The corpus contains 7842 documents that vary in length. These documents fall into ten classification categories (Arts, Computer, Economics, Education, Engineering, Law, Medicine, Politics, Religions, Sports) that vary in the number of documents. This Arabic corpus was split into the training and test sets in the ratio $3:1$.

4.3 Performance evaluation

For evaluating the performance of the technique, the typical evaluation metrics that come from information retrieval are used: *Precision* (P), *Recall* (R) and $F1$ measure [Baeza-Yates and Ribeiro-Neto 1999]:

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN}, F1 = \frac{2 * P * R}{P + R} \tag{6}$$

where TP (True Positives) are defined as the documents that were correctly assigned to the considered category while FP (False Positives) are the documents that were wrongly assigned to that category. Similarly, TN (True Negatives) were correctly not assigned to the considered category, while FN (False Negatives) were not assigned to the considered category but should have been assigned to it (since they belong to it). All presented measures can be aggregated over all categories in two ways: micro-averaging – the global calculation of measure considering all the documents as a single dataset regardless of categories, and macro-averaging – the average on measure scores of all the categories. In this article, micro-averaged F1 (mi-F1) and macro-averaged F1 (ma-F1) measures are reported.

5. EXPERIMENTAL RESULTS

One of the most important questions in the n-gram TC technique is what are the values of n and L that produce the maximum accuracy. To give an answer to this question, the accuracy (mi-F1 and ma-F1) of the technique is tested for a wide range of values of n-gram size n and the profile length L. Fig. 1 presents this extensive set of experiments for the English 20-Newsgroups corpus (the upper part of the picture) and Arabic dataset Mesleh-10 (the bottom part of the picture), for dissimilarity measure *dmod* (similar results are obtained for all other measures). In the case of 20-Newsgroups corpus we present results for the n-gram size from 6 to 8 and different values of profile length L (in the range between 50000 and 300000). In the case of Mesleh-10 corpus, we present results for the n-gram size from 9 to 11 and profile length L from 10000 to 60000 with step 5000. For all other values of n we obtained weaker results. The vertical axis indicates the accuracy, with respect to mi-F1 (the left part of the picture) and ma-F1 (the right part of the picture) in percentage, while the horizontal axis indicates the profile length L used in the TC process. It is interesting that there is a sudden drop of performance after L exceeds a certain value. This occurs when the profile length L exceeds the maximum possible profile length for at least one category, for the considered n-gram size. This is because the dissimilarity measures is affected by the size of the category profile (the bigger the profile

length, the greater the difference measure). When L became bigger then maximum possible profile length for at least one category, many documents is wrongly classified into that category. Thereby, this technique is not applicable for those (n, L) combinations where L exceeds the length of at least one category profile in the corpus.

From all results presented in the Fig. 1, we find that the performance of the classifier varies with respect to varying n-gram size. We can see that the mi-F1 and ma-F1 numbers peak at the n-gram size $n = 7$ in the case of the 20-Newsgroups corpus and $n = 10$ in the case of the Mesleh-10 corpus. For these particular values of n, empirical comparisons between measures $dmod$, d_1mod, d_2mod and d_3mod are performed. The results of these experiments for the English 20-Newsgroups corpus, for L from 50000 to 300000 are shown in the upper part of the Fig. 2 while the results for the Arabic Mesleh-10 corpus, for L from 10000 to 55000 with step 5000, are presented at the bottom part of this figure. From these results we conclude that all presented dissimilarity measures achieve similar results (maximum difference between all measures is less than 0.5%). So, the classification accuracy does not significantly depend on the choice of a dissimilarity measure. However, $dmod$ measure slightly outperforms other measures. From Fig. 1 and Fig. 2 we see that the results for mi-F1 and ma-F1 (as L increases) are quite different between English and Arabic. This can be explained by different corpora distribution over categories. The Arabic corpus is characterized by skewed frequency distribution while 20-Newsgroups is characterized by almost uniform distribution of documents over categories. Also, we see that the results for Arabic (ma-F1=92.27%, mi-F1=93.38%) are better than the results for English (ma-F1=83.1%, mi-F1=83.56%). This can be explained by double the number of categories in the English corpus compared to Arabic, with much more similar contents.

5.1 Comparison with Other Methods

In order to evaluate the technique presented in this article, the results are compared with the results obtained by other n-gram based methods and BOW SOA methods.

5.1.1 *Comparison with Other n-Gram Based Methods.* First we compare results obtained by the WnGT proposed in this article, with the results obtained by the BnGT. As opposed to the BnGT where only normalized frequencies of n-grams are used for representing each category, the WnGT takes into account not only the corresponding category, but other categories as well. Weighting factors, which are associated with n-grams in category profiles, reflect importance of n-grams for the corresponding category with respect to other categories (the smaller the number of categories an n-gram occurs in, the higher the weighting factor of that n-gram in a category it occurs in). In this way, n-gram that belongs to a smaller number of categories has a greater significance for the corresponding category. Fig. 3 shows performance comparisons between the WnGT and the BnGT, with respect to mi-F1 and ma-F1 for dissimilarity measure $dmod$ (similar results are obtained for all other measures). The upper part of this picture presents results for English 20-Newsgroups corpus for $n = 7$, and the bottom part of the figure presents results for the Arabic Mesleh-10 corpus, for $n = 10$. It can be seen that the WnGT outperforms the BnGT, for both corpora.

Now we compare our results with other published results obtained by other n-gram based techniques. In the case of 20-Newsgroups corpus, comparison is done with n-gram based TC technique presented by Rahmoun and Elberrichi [2007] in term of ma-F1 (there is no reported results for mi-F1). The best obtained result is 71.7% ma-F1 which evidence that our technique outperforms technique presented in [Rahmoun and Elberrichi 2007] (11.95% better ma-F1). In the case of Arabic Mesleh-10 corpus, there are no reported results for other n-gram based methods for this corpus to compare with.

5.1.2 *Comparison with Other SOA Methods.* In the case of 20-Newsgroups corpus, comparison is done with the reported results of the kNN and SVM presented by Lan et al. [2009] and in the case of Arabic Mesleh-10 corpus, comparison is done with the results obtained by SVM, NB, kNN and Rocchio machine learning techniques presented by Mesleh [2011]. To make the comparison more

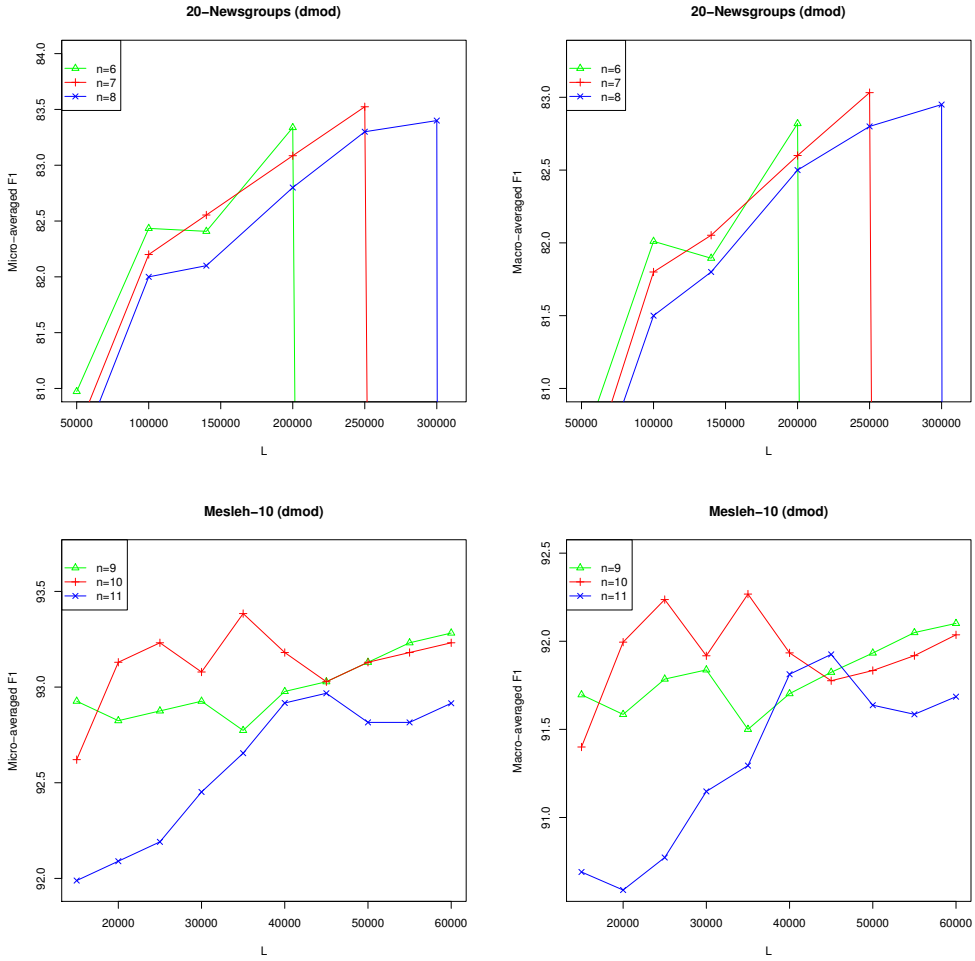

Fig. 1. Micro- and macro-averaged $F1$ for the English 20-Newsgroups and the Arabic Mesleh-10 corpora, for different values of n-gram size n and dissimilarity measure *dmod*.

convincing and fair, the same corpora and test/training split are used for all presented techniques. There is only one result for each learning algorithm, for each approach on each corpus. Therefore, only the best reported results are cited. Table I shows obtained results of all mentioned comparisons. We conclude that the WnGT outperforms SOA methods for both English and Arabic corpora.

In the case of 20-Newsgroups corpus, maximum value for ma-F1 obtained by the WnGT is 83.10% (which is 14.1% better than "kNN BOW" and 2.3% better than "SVM BOW") and maximum mi-F1 is 83.56% (which is 14.43% better than "kNN BOW" and 2.75% better than "SVM BOW"). In the case of Arabic Mesleh-10 corpus, maximum value for ma-F1 for the technique presented in this article is 92.27% which is better than ma-F1 values for all other SOA techniques (0.86% better than "SVM BOW", 4.49% better than "NB BOW", 16.46% better than "kNN BOW" and 17.55% better than "Rocchio BOW"). Maximum value for mi-F1 is 93.38%, but there are no reported results in terms of mi-F1 for this corpus to compare with.

Since SVM has been the state-of-the-art for TC for a while, for comparison purpose we conducted

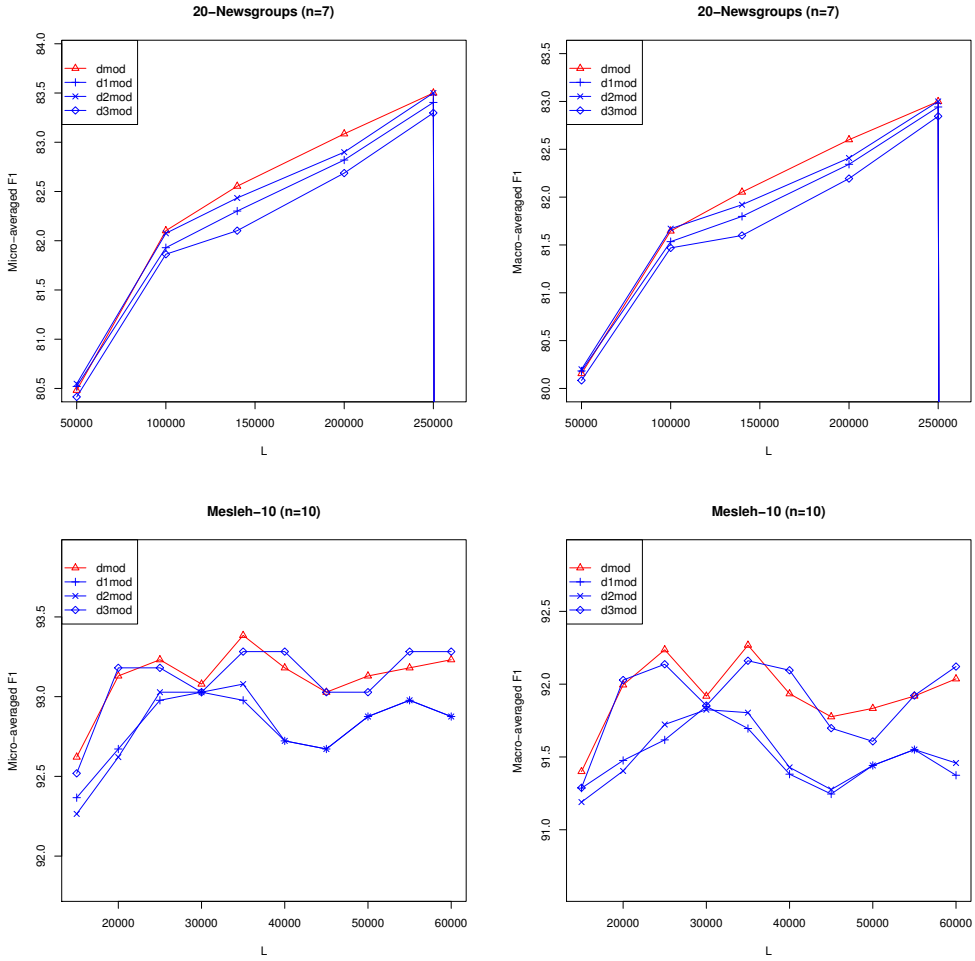

Fig. 2. Micro- and macro-averaged $F1$ for the English 20-Newsgroups and the Arabic Mesleh-10 corpora, for fixed values of n-gram size n and different dissimilarity measures.

experiments with $SVM^{multiclass}$ proposed by Joachim[2]. We used byte n-grams document representation (each document is represented by top 100, top 1000, top 5000 and all byte n-grams) and we used simple tf-idf measure: $tf(x, d)$ is the frequency of byte-n-gram x in the document d while

[2]Available at `http://www.cs.cornell.edu/people/tj/svm_light/svm_multiclass.html`

Table I. Comparison of the WnGT with other bag-of-words state-of-the-art methods.

Authors	Technique	Data collection	Ma-F1	Mi-F1
Lan et al. (2009)	SVM	20-Newsgroups	80.8%	80.81%
	kNN		69%	69.13%
Mesleh (2011)	SVM	Mesleh-10	91.41%	N/A
	NB		87.78%	N/A
	kNN		75.81%	N/A
	Rocchio		74.73%	N/A
Our proposal	kNN n-grams	20-Newsgroups	**83.10%**	**83.56%**
		Mesleh-10	**92.27%**	**93.38%**

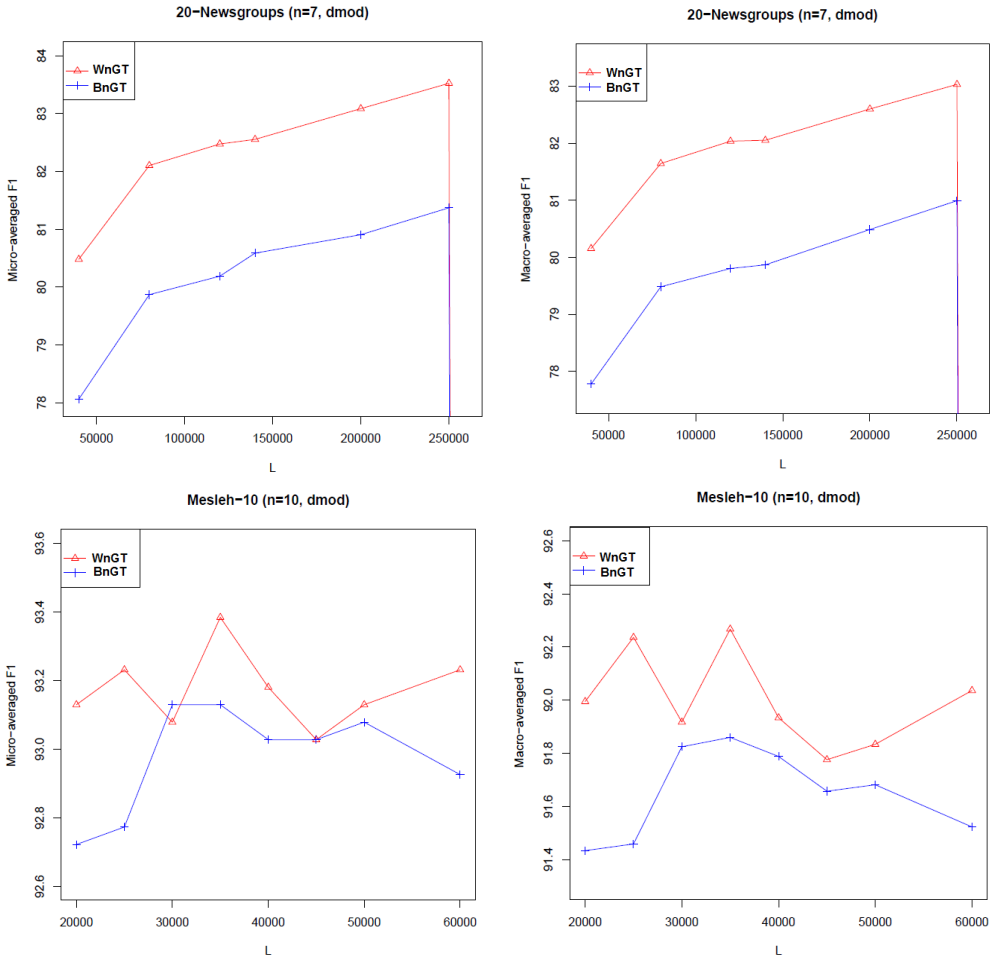

Fig. 3. Micro- and macro-averaged F1 comparison of the WnGT with the BnGT for the English and the Arabic corpora.

$idf(x) = log(\frac{N}{\{d|x \in d\}+1})$ where N is the number of documents in the corpus and $\{d|x \in d\}$ is the number of documents that contain the n-gram x. Tables II and III present obtained results for SVM for different values of parameter C (from 10^{-5} to 10^{3}). We conclude that the SVM and kNN classifiers achieve almost the same results (SVM achieve 0.19% better mi-F1 for 20-Newsgroups and 0.06% better mi-F1 for Mesleh-10 corpus) when we use byte-n-grams based document representation technique.

Table II. SVM mi-F1 results for 20-Newsgroups corpus in English.

20-Newsgroups in English				
	Top 100	Top 1000	Top 5000	All n-grams
C=0.00001	34.33	43.61	40.10	38.25
C=0.0001	37.27	45.76	43.14	43.82
C=0.001	43.48	49.24	52.29	53.18
C=0.01	54.58	67.57	69.81	68.93
C=0.1	62.01	78.12	79.06	79.25
C=1	67.71	81.03	82.12	82.04
C=10	69.89	83.52	83.71	**83.75**
C=100	71.19	83.07	83.35	83.26
C=1000	71.17	82.73	82.79	82.98

Table III. SVM mi-F1 results for Mesleh-10 corpus in Arabic.

Mesleh-10 in Arabic				
	Top 100	Top 1000	Top 5000	All n-grams
C=0.00001	79.95	82.75	82.49	82.49
C=0.0001	79.95	82.75	82.49	82.49
C=0.001	79.95	87.18	87.18	87.38
C=0.01	82.90	89.01	89.72	90.08
C=0.1	83.77	90.13	90.64	90.69
C=1	85.50	91.70	92.26	92.37
C=10	88.19	93.38	93.38	93.33
C=100	88.60	**93.44**	93.38	93.38
C=1000	88.60	**93.44**	93.38	93.38

6. CONCLUSION AND FUTURE WORK

The main contribution of the research presented is the development of a new, language independent document categorization technique based on byte-level n-grams and weighting factors. As opposed to the basic n-gram technique (BnGT) presented by Keselj et al. [2003], where only normalized frequencies of n-grams (that come from intra-document source) were used for representing each category, in this article we introduce a new weighting factors schema (that comes from inter-document source), resulting in a new n-gram technique (WnGT). The soundness of the new technique is illustrated by its successful application to the text categorization (TC) problem in two paradigmatically quite different languages – English and Arabic. Weighting factors, which are associated with n-grams in category profiles, reflect importance of n-grams for the corresponding category with respect to other categories. Experimental results confirmed our expectations that the knowledge about all categories to which an n-gram from a training document belongs can improve performance of the technique. Moreover, even without complex morphological analysis of text, this technique outperforms other n-gram based and bag-of-words state-of-the-art methods. Although optimum results for different languages are obtained for different n-gram size n and different profile length L, its overall success confirms that the WnGT is sound and promising. It provides an inexpensive and effective way of classifying documents. Since the WnGT is language independent, it is of interest to test it on corpora in other languages as well. Some preliminary experiments are already conducted on data collections in English (Reuters-21578), Chinese (Tancorp) and Serbian (Ebart-3) with very encouraging results.

The presented technique has wide application potential to different domains and problems.

Acknowledgement

The work presented has been financially supported by the Ministry of Science and Technological Development, Republic of Serbia, through Projects No. 174021 and No. III47003.

REFERENCES

Al-Diabat, M. Arabic Text Categorization Using Classification Rule Mining. *Applied Mathematical Sciences* 6 (81): 4033–4046, 2012.

Al-Harbi, S., Almuhareb, A., Al-Thubaity, A., Khorsheed, M. S., and Al-Rajeh, A. Automatic Arabic Text Classification. In *Proceedings of The 9th International Conference on the Statistical Analysis of Textual Data*. Lyon, France, pp. 77–83, 2008.

Al-Shalabi, R. and Obeidat, R. Improving KNN Arabic Text Classification with N-Grams Based Document Indexing. In *Proceedings of the 6th International Conference on Informatics and Systems INFOS*. Cairo, Egypt, pp. 108–112, 2008.

Althubaity, A., Almuhareb, A., Alharbi, S., Al-Rajeh, A., and Khorsheed, M. KACST Arabic Text Classification Project: overview and preliminary results. In *Proceedings of The 9th IBIMA conference on Information Management in Modern Organizations*. Marrakech, Morocco, pp. 1239–1244, 2008.

Baeza-Yates, R. A. and Ribeiro-Neto, B. A. *Modern Information Retrieval*. ACM Press / Addison-Wesley, Boston, MA, USA, 1999.

Dichy, J. Arabic Lexica in a Cross-Lingual Perspective. In *Proceedings of ARABIC Language Resources and Evaluation: Status and Prospects, A Post Workshop of LREC*. Las Palmas, Canary Islands, Spain, pp. 1–7, 2002.

Duwairi, R. M. Arabic Text Categorization. *International Arab Journal of Information Technology* 4 (2): 125–132, 2007.

El-Halees, A. Arabic Text Classification Using Maximum Entropy. *The Islamic University Journal* 15 (1): 157–167, 2007.

Freund, Y. and Schapire, R. E. Experiments with a New Boosting Algorithm. In *Proceedings of the 13th International Conference on Machine Learning, ICML*. Bari, Italy, pp. 148–156, 1996.

Gharib, T. F., Habib, M. B., and Fayed, Z. T. Arabic Text Classification Using Support Vector Machines. *International Journal of Computers and Their Applications* 16 (4): 192–199, 2009.

Graovac, J. A variant of n-Gram Based Language-Independent Text Categorization. *Intelligent Data Analysis* 18 (4): 677–695, 2014.

Graovac, J. and Lažetić, G. P. Language-Independent Sentiment Polarity Detection in Movie Reviews: a case study of english and spanish. In *Proceedings of the 7th ICT Innovations*. Ohrid, R. Macedonia, pp. 13–22, 2014.

Harrag, F., El-Qawasmeh, E., and Pichappan, P. Improving Arabic Text Categorization Using Decision Trees. In *Proceedings of the 1st International Conference on Networked Digital Technologies, NDT'09*. Setif, Algeria, pp. 110–115, 2009.

Ismail, H., Mahmoud, A.-A., Abdulla, N. A., Almodawar, A. A., Abooraig, R., and A.Mahyoub, N. Automatic Arabic Text Categorization: a comprehensive comparative study. *Journal of Information Science* 41 (1): 114–124, 2014.

Kanaan, G., Al-Shalabi, R., Ghwanmeh, S., and Al-Ma'adeed, H. A Comparison of Text-Classification Techniques Applied to Arabic Text. *Journal of the American Society for Information Science and Technology* 60 (9): 1836–1844, 2009.

Keselj, V., Peng, F., Cercone, N., and Thomas, C. N-gram-based Author Profiles for Authorship Attribution. In *Proceedings of the Conference on Pacific Association for Computational Linguistics, PACLING*. Halifax, Canada, pp. 255–264, 2003.

Khreisat, L. Arabic Text Classification Using N-gram Frequency Statistics: a comparative study. In *Proceedings of the Conference on Data Mining| DMIN'06*. Las Vegas, USA, pp. 78–82, 2006.

Kourdi, M. E., Bensaid, A., and eddine Rachidi, T. Automatic Arabic Document Categorization Based on the Naive Bayes Algorithm. In *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*. Stroudsburg, PA, USA, pp. 51–58, 2004.

Kursat, U. A. and Serkan, G. The Impact of Preprocessing on Text Classification. *Information Processing and Management* 50 (1): 104–112, 2014.

Lan, M., Tan, C. L., Su, J., and Lu, Y. Supervised and Traditional Term Weighting Methods for Automatic Text Categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31 (4): 721–735, 2009.

Lang, K. The 20 Newsgroups Data Set. `http://qwone.com/~jason/20Newsgroups/`, 2004.

Lewis, D. D. *Representation and Learning in Information Retrieval*. Ph.D. thesis, University of Massachusetts, USA, 1992.

Mesleh, A. Chi Square Feature Extraction Based SVMs Arabic Language Text Categorization System. *Journal of Computer Science* 3 (6): 430, 2007.

Mesleh, A. M. Feature Sub-Set Selection Metrics for Arabic Text Classification. *Pattern Recognition Letters* 32 (14): 1922–1929, 2011.

Noaman, H. M., Elmougy, S., Ghoneim, A., and Hamza, T. Naive Bayes Classifier Based Arabic Document Categorization. In *Proceedings of the The 7th International Conference on Informatics and Systems (INFOS)*. Cairo, Egypt, pp. 1–5, 2010.

Raheel, S., Dichy, J., and Hassoun, M. The Automatic Categorization of Arabic Documents by Boosting Decision Trees. In *Proceedings of the 5th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. Marrakech, Morocco, pp. 294–301, 2009.

Rahmoun, A. and Elberrichi, Z. Experimenting N-Grams in Text Categorization. *International Arab Journal of Information Technology* 4 (4): 377–385, 2007.

Saad, M. K. and Ashour, W. Arabic Text Classification Using Decision Trees. In *Proceedings of the 12th International Workshop on Computer Science and Information Technologies CSIT*. Moscow, Saint-Petersburg, Russia, pp. 75–79, 2010.

Salton, G. and Buckley, C. Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management* 24 (5): 513–523, 1988.

Sawaf, H., Zaplo, J., and Ney, H. Statistical Classification Methods for Arabic News Articles. In *Proceedings of the Arabic Natural Language Processing in ACL 2001*. Toulouse, France, 2001.

Sebastiani, F. Machine Learning in Automated Text Categorization. *ACM Computing Surveys (CSUR)* 34 (1): 1–47, 2002.

Syiam, M. M., Fayed, Z. T., and Habib, M. An Intelligent System for Arabic Text Categorization. *International Journal of Intelligent Computing and Information Sciences* 6 (1): 1–19, 2006.

Thabtah, F., Eljinini, M. A. H., Zamzeer, M., and Hadi, W. M. Naive Bayesian Based on Chi Square to Categorize Arabic Data. In *Proceedings of the 11th International Business Information Management Association Conference on Innovation and Knowledge Management in Twin Track Economies, (IBIMA)*. Cairo, Egypt, pp. 4–6, 2009.

Thabtah, F., Hadi, W. M., and Al-shammare, G. VSMs with K-Nearest Neighbour to Categorise Arabic Text Data. In *Proceedings of the World Congress on Engineering and Computer Science. WCECS*. San Francisco, USA, pp. 778–781, 2008.

Tomović, A., Janičić, P., and Kešelj, V. n-Gram-based Classification and Unsupervised Hierarchical Clustering of Genome Sequences. *Computer Methods and Programs in Biomedicine* 81 (2): 137–153, 2006.

Yang, Y. and Liu, X. A Re-examination of Text Categorization Methods. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Berkeley, CA, USA, pp. 42–49, 1999.