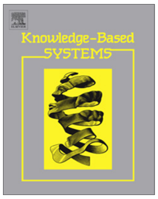
Knowledge-Based Systems 54 (2013) 288–297

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



Fuzzy unordered rule induction algorithm in text categorization on top of geometric particle swarm optimization term selection

Mustafa Karabulut [*]

Gaziantep Vocational School, Computer Programming Department, University of Gaziantep, Gaziantep, Turkey

ARTICLE INFO

Article history:
Received 8 January 2013
Received in revised form 11 September 2013
Accepted 23 September 2013
Available online 6 October 2013

Keywords:
Fuzzy rule
Geometric particle swarm optimization
Text categorization
Information gain
Multi-label text categorization
Feature selection

ABSTRACT

Rapid growth of digital information requires automated handling and organization of documents. The two main stages in automated document categorization are (i) term reduction and (ii) classification. In this paper, we present a novel two-stage term reduction strategy based on Information Gain (IG) theory and Geometric Particle Swarm Optimization (GPSO) search. We evaluate performance of the proposed term reduction approach with use of a new classifier, fuzzy unordered rule induction algorithm (FURIA) to categorize multi-label texts. In order to evaluate the performance of FURIA quantitatively, we compared it against two widely used algorithms, Naïve Bayes and Support Vector Machine (SVM). Text Categorization (TC) performance of the proposed term reduction strategy is validated with use of Reuters-21578 and OHSUMED text collection datasets. The experimental results show that performance of the proposed term reduction method is efficient for document organization tasks.

1. Introduction

Rapid growth of digital information requires automated handling and organization of documents. Mining of these documents to extract significant information is the field of Knowledge Discovery (KD) science [1]. Text Categorization (TC), an important text mining sub-field, is defined as assigning natural language text documents to predefined categories depending on the content of the documents [2]. More specifically, TC problem is defined as the task of assigning a Boolean value (*true*: document belong to the category, *false*: document does not belong to the category) to the pair $\langle d_j, c_i \rangle \in D \times C$. In this notation, $D = \{d_1, \ldots, d_n\}$ represents a domain of documents to be categorized and $C = \{c_1, \ldots, c_{|C|}\}$ corresponds to the predefined text categories for $|C| = m$.TC is used in many applications including automated document processing, populating hierarchical catalogues of Web resources and any other general application that requires selective and adaptive dispatching of documents [3].

The very first TC applications used a set of manually defined rules to encode expert knowledge for classifying documents under the given categories. Afterwards, Machine Learning (ML) algorithms gained attention for the purpose of automating TC tasks. Sebastiani [3] claims that ML based TC strategies have two main advantages: (i) categorization accuracy as good as achieved by human experts, and (ii) reduction of expert manpower, since no involvement of domain experts is required to construct a TC classifier. Some widely used ML algorithms in TC are Naïve Bayes [4], Decision Tree classifiers [5], Decision Rules [6], regression methods [7], Neural Networks [8], k-NN classifiers [9], Support Vector Machine (SVM) [10], and Rocchio classifiers [11]. In this aspect, application of novel ML algorithms to TC is an essential research field of knowledge-based systems.

TC methodologies, regardless of the preferred algorithms, are typically composed of three key components: (i) collection of texts, (ii) pre-processing of documents, and finally (iii) classifier construction to categorize the documents into pre-defined groups [13]. Unstructured texts to be categorized must first be transformed into continuous domain, i.e., numerical vectors, which are then, may be handled by ML algorithms. In other words, individual words of any text are represented in a vector space model called bag of words (BOW) [14]. This transformation is the source of major difficulty in TC tasks because native feature space (BOW vector space) consists of a great number of terms extracted from the documents such that even a moderate-sized text collection will generate tens or hundreds of thousands of terms. Such a high-dimensional feature space may cause two critical problems: (i) low TC accuracy and (ii) high computational load. In general, it is desirable to reduce required evaluation time by decreasing the size of feature space without sacrificing categorization accuracy [15]. Feature space of a TC application is, in general, reduced with two strategies: (i) removing non-informative terms according

[*] Tel.: +90 5068827506.
E-mail addresses: mkarabulut@gmail.com, mustafa_karabulut@hotmail.com

0950-7051/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.knosys.2013.09.020

to corpus statistics (feature filters or rankers), or (ii) constructing new features by combining former features (feature extractors). The most common feature reduction algorithms used in TC are the document frequency (DF), information gain (IG), mutual information (MI) and chi-square statistics (CHI) [16]. Feature rankers such as IG and CHI do not provide the subset of most prominent features (terms) automatically. Instead the user should provide the ranker algorithm with an *appropriate threshold* to obtain an optimal subset [17]. Therefore, while applying these algorithms, the number of feature words should be manually decided by the practitioners. A possible solution to this problem is the use of heuristic selection strategies such as Genetic Algorithm (GA) [18] and Particle Swarm Optimization (PSO) [28] that may generate an optimal subset of most valuable terms automatically. However, mentioned algorithms heavily suffer from the computational load problem if a large collection of terms are given. In order to overcome mentioned problems, we therefore designed a two-step feature selection strategy in which (i) firstly, a relatively smaller subset of whole collection of terms are selected with IG (e.g., selecting top 200 terms) and (ii) secondly, the best subset of terms out of these features are obtained with PSO automatically.

As a result, the first motivation of this study is to evaluate performance of the recently proposed Fuzzy Unordered Rule Induction Algorithm (FURIA) [12] for TC purposes. FURIA learns fuzzy rules on top of unordered rule sets while using a rule stretching method to deal with uncovered examples. The second motivation of this study is to evaluate the two-step feature selection methodology that combines rapidity of a ranker algorithm (the IG feature filter) with optimality and automaticity of a heuristic search procedure (PSO). In the literature, there are only few studies that make use of PSO to select features of TC application [19–22]. Therefore, an additional contribution of this paper is to study PSO with evaluation of its parameters for the purpose of combining with a ranker feature selection algorithm, IG.

We evaluated performance of the proposed strategy on Reuters-21578 [3,23] and OHSUMED [10] datasets. Additionally, we made use of frequently used TC algorithms, i.e., SVM and Naïve Bayes, to examine the performance of FURIA quantitatively. Furthermore, we made a time-complexity analysis of the proposed strategy to demonstrate its efficiency from computational load reduction point of view. The paper also contains an investigation of PSO parameters which affects the performance. Additionally, the paper presents a comparison of PSO with GA in terms of their effect on classification performance in TC tasks.

2. Preliminaries and related work

The flow of the proposed TC strategy is summarized in Fig. 1 and the details of the algorithm follow afterwards.

2.1. Benchmarking datasets

Performance of TC algorithms are evaluated with the use of benchmarking datasets. The most common TC evaluation datasets in the relevant literature are Reuters-21578 and OHSUMED. In particular, Reuters collection consists of a set of newswire stories classified under categories related to economics. Furthermore, Reuters collection comprises 21,578 documents that are arranged in 135 categories. In this study, we make use of a new split of Reuters containing 12,902 stories classified into 118 categories. The stories average about 200 words in length. For evaluation, we select eight most frequent categories (e.g., corporate, acquisitions, earnings, money, market, grain, and interest) that include a minimum of 7600 terms. The detailed distribution of the selected collection is shown in Table 1. On the other hand, it should be noted that researchers may use different splits of the same collection with different working parameters and therefore comparison about performances of TC algorithms is not a straightforward task.

We implemented the second experiment on OHSUMED dataset, a subset collection from the MEDLINE database. The database is a bibliographic catalog of important, peer-reviewed medical literature maintained by the National Library of Medicine. In this study, we consider a subset consisting of the first 20,000 documents from the 50,216 medical abstracts of the year 1991. In these documents, there are 23 Medical Subject Headings categories of cardiovascular diseases group. The subset contains 13,929 documents in 23 categories of which we make use of eight most frequent. The detailed distribution of collection is shown in Table 2.

2.2. Pre-processing steps of datasets

Automatic TC requires the documents, i.e., typically string of characters, to be converted to a scheme that ML algorithms can handle. The most common and the simplest representation is the bag of words (BOW). In this scheme, the terms occurring in a document are represented either with a binary value (simply for each term being present or not) or with a value denoting the number of times that the term occurs in the document. In this structure, each document is then represented with a *row vector* whose columns are the terms extracted from the document itself. Use of mentioned BOW strategy generates a high dimensional space, where the numbers of terms are larger than the number of samples available for training. The second problem with this strategy is the computational load. In this concept, most of the pre-processing techniques, i.e., stop token (punctuation) removal, stop word removal, stemming and term pruning serve this purpose [24]. We applied stop-token (punctuation) removal, stop-word removal, term weighting and word pruning steps on the datasets. The following sub-sections provide a basis for pre-processing steps applied to Reuters and OHSUMED datasets.

```
1: Input Reuters or OHSUMED collection
2: Tokenize the documents with Java tokenizer for delimiter set {\r,\n,\t}
3: Remove stop-tokens {.,;:'"()?!&-#0123456789+/<>$^ %[]*="}
4: Remove stop-words using SMART system word list
5: Calculate term weighting (frequency) and obtain vector space model
6: Prune terms with frequency lower than 3
7: Obtain binary vector space model (1 for term present, 0 for term absent in the document)
8: Apply IG to collection and Select top 200 terms
9: Apply GPSO search to obtain best term subset
10: Use FURIA, NB and SVM to classify documents on top of 10-folds cross-validation
11: Output TC results
```

Fig. 1. Text categorization work-flow.