Naive Rule Induction for Text Classification based on Key-phrases

Nikitas N. Karanikolas & Christos Skourlas
Department of Informatics,
Technological Educational Institute of Athens, Greece.

Abstract

In this paper, we focus on the induction of naive rules for classifying text documents. An algorithm is briefly described for the creation of key-phrases from a given set of documents and these key-phrases are organized and used as features for the automatic classification of new documents. An Authority list of key-phrases is specified by the algorithm containing key-phrases that are frequent within the documents of only one or few classes in the training set. In this framework, this last property permitted us the creation of naive rules that measure the similarity of new documents with the existing classes.

Keywords: text data mining, text classification, instance based learning, rule induction.

1 Introduction

Key-phrases or search terms could be defined as sequences of adjacent words within a text window (e.g. five successive words of the text / a sentence) forming a meaningful, descriptive phrase related to the content of the text document. Such terms can be used as features for classifying (text) documents. Since, not every key-phrase is appropriate for discriminating between documents, we have to examine and apply methods for selecting the appropriate ones. Hence, a prerequisite for such a classification method is the use and maintenance of a list of key-phrases, the so-called "Authority List" Karanikolas et al [4]. An interesting problem is related to the reduction of the search space that is needed for the extraction of candidate key-phrases.

In Classification learning, a learning scheme takes a set of classified examples from which it is expected to learn a way of classifying unseen examples. In Instance-based learning methods we start with a particular instance (example) and see how it can be generalized to cover other nearby instances (examples) in the same class Witten et al [10]. Therefore, Instance based learning methods seems to be a novel way for classifying documents. In such a method a similarity measure adapted from the domain of Text Retrieval (Kowalski [7], Deal [1]) can be used to evaluate how close to the existing classified documents is a new, under classification, document.

1.1 Definitions - Notations

Every classified document could be represented (or accompanied) by a vector of (m+1) items. The first m items are assigned to Boolean values, representing the existence or not of a corresponding key-phrase in the document. It means that the elements of the "Authority" list of key-phrases are m or in other words that the number of key-phrases selected for classification is m. In the last (m+1)-item of the vector is located the class, called the label, of the document. Having a training set of labeled vectors we eventually form a table with m+1 attributes (columns). Thus, data mining algorithms (or knowledge discovery) can be easily applied to such a representation of documents, in order to extract rules or trees for classifying new documents Karanikolas et al [4].

2 Materials and Methods

In the proposed classification scheme we use an authority list of key-phrases and a set of classified documents and try to apply a similarity measure which offers a way of classifying new (unclassified) documents. Hence, in our work, Text Classification is the task of applying (semi) automatic methods in order to select the appropriate class for a given document, between a set of predefined ones (classes). More precisely, our instance based learning approach for the classification of documents is based on the use of the key-phrases and the related concept of the similarity between existing documents (the "training" set) and the new (under classification) documents Karanikolas et al [5].

The method assumes that similar documents must be classified in the same category, i.e. must share the same classification code. It remains to select the features of documents that must be used for measuring their similarity. In our approach these features are key-phrases that exist in the documents of the training set and are extracted from the Authority List. The representation of our documents is a vector with (m+2) attributes. The first attribute contains the document. Each of the next m attributes is assigned to a Boolean value that represents the existence or not of a corresponding key-phrase in the document. The last item of the vector is the classification code (the label) of document. The following figure (fig. 1) depicts the vector and its relationship with the list of key-phrases and the list of available classes.

This approach exploits a well-known approach in the domain of Information Retrieval in order to define a measure of similarity between the new document and the documents of the training set. Our proposal for a similarity measure is an adaptation of the cosine measure (see for example Lucarella [8]).

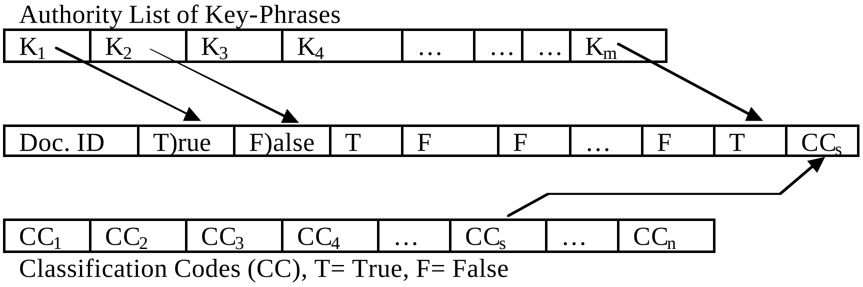

Figure 1: The representation of our documents' vectors and its relationship with the list of key-phrases and the list of available classes

The following equation calculates the similarity:

$$S(D_i, D_{new}) = \frac{\sum_{j=1}^{m} q_j k_{ij}}{\sqrt{\sum_{j=1}^{m} q_j^2 \cdot \sum_{j=1}^{m} k_{ij}^2}} = \frac{\sum_{j=1}^{m} q_j k_{ij}}{L_{D_{new}} \cdot L_{D_i}} \qquad (1)$$

where *m* is the number of key-phrases used in the collection, k_{ij} is equal to 1 if the key-phrase j exists in document D_i (of the training set), otherwise is equal to 0 and q_j is the weight of key-phrase j in the new document. The following equation can be used to measure the term q_j:

$$q_j = \log_2\left(\frac{ClassCount}{ClassFreq_j}\right)$$

where *ClassCount* is the number of classes of the training set, and $ClassFreq_j$ is the number of classes that include the key-phrase j.

Selecting key-phrases (Georgantopoulos et al [3], Turney [9]) that will be used as attributes for text classification is a very important step in our approach. The choice of the key-phrases has not to be based on frequent (within the whole text collection) candidate phrases. We must also avoid choosing key-phrases that exist in a few texts of the collection (as a whole) but are quite frequent within these texts Karanikolas et al [6]. The creation of an Authority List and its use for document classification must be based on the selection of key-phrases that are frequent within the documents of only one or few classes in the training set.

2.1 Phrase Extraction

We formalize the problem of phrase extraction for classification in the following way. Given a collection of documents subdivided into classes, a window width and a frequency threshold, find all key-phrases that occur frequently enough in only one or few classes. The following algorithm can solve the problem of extracting the appropriate key-phrases and has two alternating phases: building new candidate key-phrases, and evaluating how often these ones occur in a class of the collection.

Algorithm

```
 1 For every class ($CL_i$) of the training set do
 2    For every document of the class ($DCL_i$) do
 3       Stemming
 4       stop word removal
 5    End {For every document of the class}
 6    Choose the most frequent stems of the class (P0 - parameter)
 7    Form the candidate double word phrases ($C_2$) from the
      frequent stems ($L_1$)
 8    Choose the most frequent double word phrases ($L_2$)
      (W1 and P1 - parameters)
 9    For x=3,4 do
10       Form the candidate x - width word phrases ($C_x$) from
         the frequent (x-1) – width word phrases ($L_{x-1}$)
11       Choose the most frequent x – width word phrases ($L_x$)
         (P2 and W2, P3 and W3 – parameters)
12    End {For x=3,4 do}
13    Compose an integrated list by joining $L_x$ (for x=2,3,4).
      This join, forms the frequent word phrases of class ($FCL_i$)
14 End {For every class of the training set}
15 Integrate / Join the lists of frequent word phrases of all
   classes of the training set
16 Reject the frequent word phrases that exist in many classes
   (Pt - parameter). The rest of the frequent word phrases form
   the set of key-phrases or «Authority list»
17 Form the dictionary of «Terms». It is the list of stems that
   are components of the key-phrases of the «Authority list».
```

Where Parameter *P0* is the percentage of texts of the class that must contain a stem, *W1* is the width of window that covers 2-word phrases, *P1* is the percentage of texts of the class that must contain a 2-words phrase, etc. *Pt* is the percentage of classes that can contain a key-phrase.

The way that step10 forms the candidate x-width word phrases (C_x) from the frequent (x-1) – width word phrases (L_{x-1}) is an interesting part of our algorithm. The efficiency of the algorithm is based on the following estimation: *Potentially, a very large number of candidate key-phrases has to be checked. Hence, we can reduce the search space by building larger key-phrases from smaller ones. In* *other words, it is only necessary to test the occurrences of key-phrases whose all sub-(key-phrases) are frequent.*

3 Naive Rules induction

In this section we focus on some key points for the induction of naive rules that measure the similarity of a new, unclassified, document with the pre-defined classes of documents.

There is an interesting property incorporated into the above Key-phrase Extraction algorithm: The Authority list created by the algorithm contains key-phrases that are frequent within the documents of only one or few classes. Consequently, for each class we can have a list of "representative" key-phrases. A framework for extracting naive rules can be the following one: The similarity of a new document with the elements of some class CL_i is estimated by the number of key-phrases from the list of the representative ones for the class CL_i. More precisely, the calculation of similarity is equal to the number of items of CL_i class' representative key-phrases list that actually exist in the new document, divided (normalized) by the population of the list.

The following notation denotes the list of representative key-phrases of class CL_i: $RCL_i = \{kpCL_{i1}, kpCL_{i2}, ..., kpCL_{ir}\}$, where $kpCL_{ij}$ is the j-(key-phrase) of the representative key-phrases of class CL_i. The measure $|RCL_i|$ defines the number of the representative key-phrases for the specified class CL_i.

A simple framework for extracting naive rules that measure the similarity of a new, unclassified, document with the class CL_i is defined by the equation:

$$S^{'}(D_{new}, CL_i) = \frac{count(exist(kpCL_{ij}, D_{new}))}{|RCL_i|} \tag{2a}$$

where the operation (function) "exists" denotes if a representative key-phrase $kpCL_{ij}$ exists in the new document and "counts" means the calculation of the existing key-phrases.

If we want to proceed inducting more complicated rules we can use weights based on the number of words (stems) that constitute a key-phrase.

$$S^{'}(D_{new}, CL_i) = \frac{\sum_{j=1}^{|RCL_i|} count(contituent(exist(kpCL_{ij}, D_{new})))}{\sum_{j=1}^{|RCL_i|} count(constituent(kpCL_{ij}))} \tag{2b}$$

Thus, given a new document we can measure its similarity with every class, using formula (2a) or (2b), and suggest the most plausible classes, i.e. the classes that have the greater similarity with the new document. In other words, we have to apply the similarity measures $S^{'}(D_{new}, CL_i)$ for any available class CL_i and choose the class that gets the highest value. On the contrary, an instance based learning approach needs the application of the formula for the calculation of the similarity $S(D_i, D_{new})$ as many times as the size of the training set. Moreover, in order to select the most promising class, we have to apply a second step that finds the “average” similarity of the new document with the documents of each class of the training set. This can be obtained with the following formula:

$$S^{''}(D_{new}, CL_i) = \frac{\sum_{D_j \in DCL_i} S(D_j, D_{new})}{\left|DCL_i\right|} \tag{3}$$

where DCL_i is the subset of the training’s set documents that are pre-classified as members of class CL_i.

Example

Let say that we have a training set that contains a thousand (1000) documents pre-classified in eight (8) classes. In order to select the most relevant class for a new document, with the usage of the instance based learning approach, we have to:

– apply formula (1), between every document of the training set and the new unclassified document, i.e. a thousand of times

– apply formula (3), to get the average similarity between each class and the new unclassified document, i.e. eight times

On the contrary, the application of naive rule (2a), eight (8) times is the only step needed to select the most relevant class for the new document.

Thus, the naive rule induction based approach is simple and could be easily applied in various applications. On the other hand, the instance based learning approach can be applied only when the training set is not a very extended one.

4 Conclusions

We have presented methods for (semi) automatic classification of documents based on instance based learning and naive rule induction. Our approach is based on the use of key-phrases from a controlled list (Authority List).

The creation and maintenance of the key-phrase’s (authority) list is an important subject in the domain of document classification. Our algorithm for key-phrase selection permits the induction of naive rules for classification. There is an interesting property in our algorithm, for selecting the appropriate key-phrases for classification that permitted us to investigate a naive rule induction for classification, as an alternative to the classical data mining algorithms. The Authority list is created by the proposed algorithm and contains key-phrases that are frequent within the documents of only one or few classes in the training set. Consequently, for each class we can have a list of “representative” key-phrases.

A framework for naive rules’ induction is based on a measure of the similarity between a new document and the representative key-phrases of some class $CL_{i.}$ Hence, if a new document is given we can measure its similarity with every class and suggest the most plausible classes, i.e. the classes that have the greater similarity with the new document. It is also possible to extract more complicated rules using weights e.g. weights based on the number of words (stems) that constitute a key-phrase.

Acknowledgements

This work was co-funded by 75% from E.E. and 25% from the Greek Government under the framework of the Education and Initial Vocational Training Program – Archimedes.

References

[1] Deal D.C., Techniques of Document Management: A review of Text Retrieval and related technologies. *Journal of Documentation*, **57**, pp. 192-217, 2001.

[2] Frank Eibe, et al. Domain-Specific Key-phrase Extraction. *International Joint Conference of Artificial Intelligence*, 1999.

[3] Georgantopoulos B. & Piperidis S. Automatic Term Extraction Based on Pattern Grammars. *LogoNavigation*, Issue 5, May 1999.

[4] Karanikolas N.N. & Skourlas C. Automatic Diagnosis Classification of patient discharge letters. *MIE'2002: XVIIth International Congress of the European Federation for Medical Informatics,* Budapest, August, 2002. IOS Press, ISBN: 1-58603-279-8, ISSN: 0926-9630.

[5] Karanikolas N.N., Skourlas C., Christopoulou A. & Alevizos T. Medical Text Classification based on Text Retrieval techniques. *MEDINF 2003. 1st International Conference on Medical Informatics & Engineering*, October 9 - 11, 2003, Craiova, Romania.

[6] Karanikolas N.N. & Skourlas C. Key-Phrase Extraction for Classification. *MEDICON and HEALTH TELEMATICS 2004. X Mediterranean Conference on Medical and Biological Engineering,* July 31 - August 5, 2004, Ischia, Italy.

[7] Kowalski G. *Information Retrieval Systems. Theory and Implementation,* Kluwer Academic Publishers, 1st edn, 1997, ISBN 0-7923-9899-8.

[8] Lucarella, D., A document retrieval system based on nearest neighbour searching, *Journal of Information Science,* **14**, pp. 25-33, 1988.

[9] Turney P. Extraction of Key-phrases from Text: Evaluation of Four Algorithms. *National Research Council of Canada*, Technical Report ERB-1051, October 23, 1997.

[10] Witten Ian & Frank Eibe, *Data Mining: Practical Machine Learning tools and Techniques with Java implementation*, Morgan Kaufmann, 1999, ISBN: 1-55860-552-5.