The Journal of Systems and Software 82 (2009) 1036–1045

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Using phrases as features in email classification ☆

Matthew Chang, Chung Keung Poon *

Department of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon Tong, Hong Kong, China

ARTICLE INFO

Article history:
Received 25 March 2006
Received in revised form 14 January 2009
Accepted 14 January 2009
Available online 22 January 2009

Keywords:
Document classification
Email
Resemblance
Nearest-neighbour
Naive Bayes

ABSTRACT

In this paper, we report our experience on the use of phrases as basic features in the email classification problem. We performed extensive empirical evaluation using our large email collections and tested with three text classification algorithms, namely, a naive Bayes classifier and two *k*-NN classifiers using *TF–IDF weighting* and *resemblance* respectively. The investigation includes studies on the effect of phrase size, the size of local and global sampling, the neighbourhood size, and various methods to improve the classification accuracy. We determined suitable settings for various parameters of the classifiers and performed a comparison among the classifiers with their best settings. Our result shows that no classifier dominates the others in terms of classification accuracy. Also, we made a number of observations on the special characteristics of emails. In particular, we observed that public emails are easier to classify than private ones.

1. Introduction

Rapid growth of the Internet has led to a proliferation of emails. Nowadays, it is common for an email user to receive tens or even hundreds of emails everyday. To organize our emails so that they can be searched and maintained efficiently, we often group them into files. However, reading the emails one by one and filing them by hand is still a tedious process. Moreover, the problem is getting worse as the number of emails and folders keep increasing. Thus, the problem of automatic email classification is important and has gained much attention, especially in recent years. In this paper, we study the problem and focus on the use of phrases as basic features of the emails. In this section, we will present some important related research works, our contributions for email classification, and the organization of this paper.

1.1. Previous works

In the brief review below, we group the previous works in email classification into three main categories, namely, TF–IDF, statistical and rule-based classifiers.

In the TF–IDF approach (Salton (1991)), each email is mapped to a vector based on the term frequency (TF) and inverse document frequency (IDF) of each keyword presents in the email collection. Classification is then done by algorithms such as *k*-means, *k*-nearest neighbour (*k*-NN) or support vector machines (SVM). Systems following this approach and using the *k*-means algorithm include MailCat (Segal and Kephart, 1999) and the system of Manco et al. (2002). A variant of the *k*-NN algorithm, called Ibpl1, has been used as one of the core learning algorithms in the Magi system of Payne and Edwards (1997).

A simple and yet powerful statistical method is the naive Bayes classifier. In this method, each class of emails is modelled as a probability distribution of keywords, again, based on keyword frequencies; and each email in a class is assumed to be generated by drawing words randomly and independently from that distribution. Classification is done by finding the class that maximizes the probability of generating the email in question. Such a classifier has been implemented in the ifile system of Rennie (2000). Brutlag and Meek (2000) compared the performance of *k*-means, SVMs and naive Bayes classifier. They found that different datasets caused more variations in the classification accuracy than different classification algorithms.

In contrast to the two previous approaches which assign fractional values to keywords in the classifiers, rule-based approach resulted in classification rules that have discrete values (often zero-one values) on keywords and appear to be more human-readable. The Ishmail system of Helfman and Isbell (1995) allows users to specify keywords or phrases to be included or excluded. However, constructing classification rules by hand is cognitively demanding and therefore the applications of various automatic rule-learning algorithms have been investigated. These include the Ripper algorithm of Cohen (1995) studied in Cohen (1996), the cn2 algorithm of Clark and Niblett (1989) studied in Payne and Edwards (1997), the id3 algorithm of Quinlan (1986) studied in Crawford et al.

☆ The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong SAR, China [RGC CityU 1198/03E].

* Corresponding author. Tel.: +852 2788 7157; fax: +852 2788 8614.
E-mail addresses: kcmchang@cs.cityu.edu.hk (M. Chang), ckpoon@cs.cityu.edu.hk (C.K. Poon).

0164-1212/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2009.01.013

(2001) and the association rule algorithms investigated in Itskevitch (2001). Some of them are found to be quite competitive compared with the traditional TF–IDF-based algorithms, see (Cohen, 1996; Payne and Edwards, 1997).

Common to all these works, except Itskevitch (2001), is that they only consider individual keywords as the basic features to be learnt by the classifiers. The association rule algorithms of Itskevitch (2001) do consider the co-existence of keywords in an email but ignore the proximity relationship between keywords. On the other hand, our recent work (Poon and Chang, 2003) has investigated a *k*-NN classifier that is based on the concept of *shingling* and *resemblance* (Broder, 1997). Briefly, the shingling of an email captures all possible *phrases* of a certain length present in that email while resemblance is a similarity measure defined on the shinglings. Resemblance, also called Jaccard similarity coefficient (JSC) (Jaccard, 1912), and Dice similarity coefficient (DSC) Dice, 1945), which is similar to resemblance, are typically used as a measure of overlaps between sets. They range from 0, indicating no similarity between two sets, to 1, indicating complete agreement.

Previous studies on the use of phrase in text categorization have considered several different definitions of phrases. *Syntactic* phrases are based on a grammar of the language. Acquiring the relevant phrases and contexts often require complex natural language processing (Riloff and Lehnert, 1994). Moreover, experimental evaluation (e.g., Dumais et al., 1998; Koster and Seutter, 2003) so far has not been very encouraging. Our notion of phrases is closer to that of *statistical* phrases in which a sequence of words is deemed to be a phrase if such sequence occurs, say, at least three times in the collection. The purpose of such or other statistical filters are to curb the feature space to a manageable size as well as to avoid over-fitting (Caropreso et al., 2001). Usually, these techniques consist of scoring each feature by means of a *feature evaluation function* (FEF) and then selecting the subset of features with the highest scores. Many functions, mostly from the tradition of decision or information theory, have been used as FEFs in text classification, see (Lam and Lee, 1999; Mladenic, 1998; Yang and Pedersen, 1997). Some positive results have been shown in Tzeras and Hartmann (1993), Mladenic and Grobelnik (1998), and Caropreso et al. (2001). However, having such ad-hoc threshold values in a definition is not too satisfactory as the appropriate threshold values may vary according to different situations. Also, it requires significant computation time to generate the "valuable" *n*-grams. Hence, we will use an efficient and simple sampling method to limit the feature space.

1.2. Our contributions

1.2.1. Applicability of shingling

While our preliminary work (Poon and Chang, 2003) shows that phrases are effective in a *k*-NN-based email classifier, it is not known if the same is true for other classifiers. In this paper, we performed a more thorough evaluation of that classifier and investigated the applicability of shingling to several email classifiers. The classifiers include a naive Bayes classifier, and two *k*-NN classifiers with resemblance measure and cosine measure respectively.

Our notion of phrases and computation techniques are different from what have been studied before in text categorization. We consider phrases of a fixed size rather than a mixture of keywords and phrases. This is different from most other works, including (Mladenic and Grobelnik, 1998; Johannes, 1998; Caropreso et al., 2001), which considered a mixture of phrases of different sizes. Also, in our experiment, the features we consider are word-based *n*-grams which are clearly different from character-based *n*-grams.

As we define phrases as *all* sequences of words of a certain length, we leave the efficient computation of such phrases (or a suitable subset of it) as a separate problem. Such definition is more desirable because in different situations, different computations may be needed and yet our definition of phrases remains impartial to these differences. For example, as the readers will see in our experimental result, no feature selection is needed for the naive Bayes classifier while randomized feature selection technique is applied for the *k*-NN classifiers. The details of the notion and techniques will be shown in Section 2.

To our best knowledge, this is the first report on email classification problem that is based on a notion of *phrases*. Our study here indicates that using phrases of size 2 generally gives better results.

1.2.2. Feature selection

A big research focus is on limiting the feature space because of the huge number of distinct terms. In principle it would be best to identify features with concepts the document deals with, or with the *problems* the document tackles. However, extracting these pieces of knowledge are not within the current knowledge extraction technology.

To deal with the problems, we study the use of a randomized feature selection technique. More precisely, in that randomized computation, we apply a special *coordinated sampling* to achieve feature selection. Although it was shown in Theorem 1 of Broder (1997) that such a sampling method will preserve the *resemblance* (to be defined in Section 3.1) between any pair of shinglings, one of our goals in this investigation is to find out, experimentally, if such sampling preserves other similarity measures as well. Also, we will highlight the contribution of coordinated sampling experimentally by comparing with uncoordinated sampling.

1.2.3. Thorough investigation in email classification

In our experiments, we made a number of investigations on the parameters of email classifiers. The results may be useful as some guidelines and suggestions for developing an email classifier. Also, it may provide insights for the future researches on the same area.

The classifiers we have chosen are suitable for the email domain which requires constant updating of the classifiers. We describe an interactive and incremental classification model (in Section 4.2) which is a realistic model for email classification. Also, we employ this model for all our experiments. Of course, we cannot preclude the existence of other text classifiers suitable for emails. Our purpose here is not to exhaust all text classifiers that can be applied to email classification but to show that phrase featuring is useful for at least some of them. In this paper, we choose two traditional text classifiers, namely, the naive Bayes classifier and the *k*-NN classifier for investigation. Also, we will show some interesting observations of email classification including the special noise of emails that cannot be observed in traditional text classification.

To begin with, we will explain the concept of shingling and the related techniques for reducing the computational overhead in Section 2. In Section 3, we justify our choice of classification algorithms and describe the adaptation when shingles are used as the basic features. Implementation details are explained in Section 4. Then we describe our datasets and present our experimental results in Section 5. A number of interesting observations, some specific to emails, will be made. Finally, we conclude and discuss our findings in Section 6.

2. Shingling

In this section, we treat each email (or more generally, each document) as a sequence of words and explain the concept of *w*-shingling in Section 2.1. Techniques for reducing the computational overhead are then described in Section 2.2.