CSLT TECHNICAL REPORT-20160009 [Thursday 8th December, 2016]

Highly Restricted Keyword Selection Based on Sparse Analysis for Uyghur Text Categorization

Dong Wang[1,2*], Askar Humdulla[3], Rayilam Parhat[1,3] and Javier Tejedor[4]

[*]Correspondence: wangdong99@mails.tsinghua.edu.cn
[1]Center for Speech and Language Technology, Research Institute of Information Technology, Tsinghua University, ROOM 1-303, BLDG FIT, 100084 Beijing, China
[2]Center for Speech and Language Technologies, Division of Technical Innovation and Development, Tsinghua National Laboratory for Information Science and Technology, ROOM 1-303, BLDG FIT, 100084 Beijing, China
Full list of author information is available at the end of the article

Abstract

Text categorization (TC) has achieved significant success in recently years; however, in the case where the text is not well represented, TC performance is usually substantially reduced. A particular example of such a scenario is in the content-aware public telephone network (PTN), where the input speech can be only partially transcribed due to the concern of privacy protection and computational cost. One, therefore, needs an effective approach to selecting a highly restricted group of keywords (less than 100), by which the spoken content can be well represented and so the TC performance is largely retained.

Conventional keyword selection approaches are based on a carefully designed intermediate score, and the keywords are selected according to the score independently. This often leads to suboptimum performance. This paper proposes a novel sparsity-based approach to tackling the highly restricted keyword selection for TC. The idea is to formulate keyword selection as an l_1 regularized linear optimization problem. The l_1 term drives less important dimensions of the model coefficients to zeros, and so the corresponding words are nullified, leaving only the promising keywords. By this approach, the objective function of keyword selection is more consistent to the one used in TC; more importantly, the keywords are selected jointly as a group, leading to a group-optimized selection. The experiments conducted on an Uyghur TC task demonstrated that the proposed approach is highly effective.

Keywords: sparse discriminative analysis; text classification; keyword selection

1 Introduction

Text categorization (TC) has gained much attention in the research community and found numerous applications in information retrieval (IR) and text data mining. Conventional TC works on pure text where the lexicon is large (maybe unlimited). In this paper, we focus on a particular TC task where the lexicon is highly restricted (less than 100 words). An example of such text is “np np browse np np np np search”, where “browse” and “search” are in-lexicon words (keywords), and “np” is a trivial filler token that represents an occurrence of an out-of-lexicon word. This type of TC is highly desirable for spoken content search, for instance in a content-aware PTN system described as follows.

1.1 TC for content-aware PTN

The public telephone network (PTN) is a major channel for people to transmit messages. Currently, most of the PTNs are content-unaware, which means that the speech content transmitted through the PTN is undiscovered unless this is deliberated checked by authorized listeners. Comparing to the content-unaware PTN, a content-aware PTN is more desirable for multiple purposes such as quality check and information distillation. For example, the increasing concern on public security leads to resurgent requests for monitoring PTNs and detecting malicious conversations such as the criminal or violent ones; however, monitoring PTNs by human listeners is unaffordable in cost; more importantly, it violates laws of private information protection. An automatic monitoring system is obviously a much better solution. In fact, the current automatic speech recognition (ASR) technique [1, 2] has been able to deliver a reasonable accuracy on telephone speech, and with a conventional TC backend, it is feasible to deliver a content-aware PTN system.

This ASR+TC approach, unfortunately, is almost unacceptable. Firstly, nobody would like his/her conversations being recorded and transcribed, even by machines. Full transcribing is risky in private information exposition and thus usually violates national laws. Secondly, the volume of speech data transmitted by PTNs is huge, which makes full transcribing unaffordable due to the cost on memory and CPUs. Thirdly, only a very small proportion of conversations through PTNs contain malicious contents, and so it is not economic to conduct the heavy-loaded full transcribing for such a small proportion.

A reasonable approach is to select a very small group of keywords (e.g., less than 100) to monitor. Instead of conducting full transcribing, the ASR system just detects occurrences of the keywords (the other words are recognized as trivial fillers 'np'). This 'partial ASR' is also named as keyword spotting, and the recognition result is a special form of 'partially transcribed spoken text'. A TC component is then applied to the spoken text to detect the desired information, e.g., malicious conversations. Note that the partially transcribed text only contains keywords and their frequencies, therefore largely preventing information exposition. In addition, the partial transcribing is much cheaper than the full transcribing, leading to a light-weighted content-aware PTN system.

A key issue of this keyword-based content-aware PTN system is how to select the highly restricted keywords. Usually, a TC system requires thousands or tens of thousands words to form a reasonable text representation (feature vector). The partially transcribed spoken text, however, is a limited representation of the speech content due to the restricted words, which plays a big challenge for TC. It is, therefore, highly important to select the most discriminative words to ensure that the TC performance is not degraded much with the limited text representation, which is the central work of this paper.

Fig. 1 illustrates the architecture of the partial ASR-based content-aware PTN system, which involves three main components: keyword selection, partial ASR, and TC. The keyword selection component determines a group of keywords to monitor; the partial ASR transcribes speech signals into partially transcribed spoken text; and the TC component classifies the transcribed spoken text. The focus of this paper is the keyword selection component, as indicated by the shadow in Fig. 1.

In the rest of the paper, we choose the support vector machine (SVM) [3, 4] as the TC classifier. For the partial ASR component, we do not involve a real ASR system; instead, we simulate its function by replacing all the non-keywords with the trivial filler 'np'. This simulation certainly ignores some difficulties in a real ASR-based system such as false detections and missings. However, since our focus

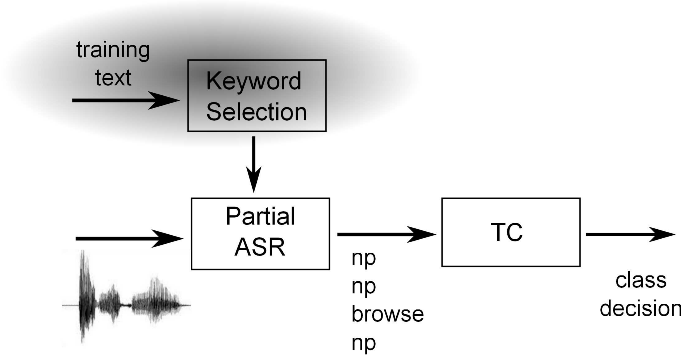


Figure 1 Architecture of a content-aware PTN system.

is the keyword selection, we believe that this simplification is reasonable and help us concentrate on the most interesting component.

1.2 Keyword selection for TC

Keyword selection has been studied in TC as an important approach to feature dimension reduction. The most commonly used keyword selection method defines an intermediate score and then selects the keywords based on the score. Some widely studied intermediate scores involve Gini index [5, 6, 7], information gain (IG) [8, 9], mutual information (MI) [10, 11], χ^2 test [12], class discriminating measure (CDM) [13], weight of evidence for text [14, 15], odds ratio and its variants [16, 15, 13], and expected cross entropy [17].

Keyword selection is also studied in information retrieval (IR). Different from the methods in TC where the focus is the discriminative capability of the selected keywords, the IR-based approach focuses on word representativeness. For instance, the selection approaches based on document frequency (DF) [18], term strength (T-S) [19, 20], and word salience [21] belong to this category. This category also involves the graph-based keyword selection, for instance the TextRank algorithm [22, 23], where the representative capability of a word is represented by the degree that it connects to other words (more details in Section 2.2).

All the above approaches can be regarded as examples of the 'filter-based approach' according to [15]. The main advantage of this type of approaches is its simplicity. However, as the selection criterion (IG, MI, etc.) is different from the objective function of the TC classifier (e.g., SVM), the selected keywords are not necessarily optimal for the TC task. Moreover, since the keywords are selected individually and independently, the selected keywords are not necessarily optimal in the sense of a group.

The 'wrapper-based approach' may partly solve these problems. Different from the filter-based approach, the wrapper-based approach selects a group of keywords simultaneously, and the selection is based on the same learning algorithm and evaluation metric that are used to learn and evaluate the TC system [15]. However, this approach requires building full systems for a large number of candidates of the keyword group, thus being very costly. In addition, searching for the optimal keyword group often relies on random walk, genetic algorithms or heuristic rules [24], which often leads to suboptimum performance. For these reasons, the wrapper-based approach is seldom used in TC and IR.

1.3 Motivation of the paper

In this paper, we propose a novel keyword selection approach based on sparse analysis. The basic idea is to train an l_1 regularized linear model $y(x) = w^T x + \lambda ||w||_1$, where x is the feature vector of an input text and each dimension of x corresponds to a word, and y is the category assignment score of x. w is the model parameter and λ is a tunable hyperparameter. Due to the l_1 constraint on w, some dimensions of w will be driven to zeros during the training process, leading to a natural way for keyword selection. A desirable feature of this approach is that multiple keywords are selected in a group-optimal way. Note that this group-optimization is a consequence of the sparse constraints. This is fundamentally different from the 'trial and error' method in the wrapper-based approach, and so is much more solid in theory and efficient in practice. When training the keyword selection model, we choose an objective function which is the same as or related to the one used in training the TC model (which is an SVM in our work), and so the keyword selection is highly consistent to the TC in terms of optimization criteria.

Two sparse models are studied in this paper: a sparse discriminative analysis (SDA) model which is a sparse version of the conventional linear discriminative analysis (LDA), and a sparse support vector machine (sparse SVM) model which is also linear but uses maximum margin as the objective function. As mentioned before, the SVM is used as the TC classifier in our work due to its excellent performance in TC [3, 4]. For this reason, the sparse SVM-based keyword selection tends to be more consistent with the TC component (both are based on maximum margin). Nevertheless, the SDA enjoys the property of simplicity in model training.

The rest of the paper is organized as follows: Section 2 reviews some related work. Section 3 presents the sparsity-based keyword selection. Section 4 reports the experiments with an Uyghur database, and the paper is concluded in Section 5.

2 Related work

This work is related to a multitude of research including spoken document retrieval, information retrieval, text categorization, and keyword extraction. In this section, we review some work on text categorization and keyword extraction which we find directly relevant to our proposal.

2.1 Text categorization

The problem of TC is to classify an input text into a number of predefined categories [25, 26, 27, 28]. TC has found a wide range of applications in text mining and information retrieval (e.g., news filtering, document organization, opinion mining, e-mail classification, and spam filtering). The content-aware PTN is a new application of TC, where the input to classify is a partially transcribed spoken text.

A typical TC system first extracts some features from the training text and builds a classifier with the features. The category of a new input is then predicted by the classifier. Many classifiers have been studied in TC, including (sorted by model complexity) Rocchio's algorithm [11], k-nearest neighbors (k-NN) [29], naive Bayes

(NB) [13], decision trees (DT) [30], neural networks (NN) [31], and SVMs [3, 4]. It has been confirmed that the SVM model is highly effective for the TC task; we therefore choose the SVM as the classifier in this paper. For more comparison among the classifiers, please refer to [27, 28].

Most of the commonly used features in TC are based on words, which usually results into a very large feature vector. Therefore, it is essentially important to reduce the dimensionality of the features so that a robust classifier can be trained. Low dimensional features are also important to speed up model training and inference. The main work of this paper falls on this research area.

There is a multitude of research on feature dimension reduction for TC. A common approach defines an intermediate score to represent the 'importance' of a word. A group of words is then independently selected according to the score, and the feature vectors are hence shortened by considering the selected words only. Examples of the intermediate scores involve document frequency [18], term strength [19, 20], and word salience [21]. Another group of intermediate scores is more related to the discriminative power of a word, including Gini index [5, 6, 7], information gain [8, 9], mutual information [10, 11], χ^2 test [12], class discriminating measure [13], weight of evidence for text [14, 15], odds ratio and its variants [16, 15, 13], and expected cross entropy [17]. Several comparative studies have been conducted to compare these scores. [18] found that DF, IG, and χ^2 are among the best and they are highly correlated. [7] reported good performance with Gini index, but the difference between Gini index, IG, and MI is rather marginal.

An obvious disadvantage of the intermediate score approach is that the words are selected individually and independently, which does not guarantee an optimal selection for the keywords as a whole group. In addition, the choice according to an intermediate score is not necessarily optimal for TC due to the discrepancy between the criteria of the word selection and the TC classifier. The sparsity-based approach presented in this paper solves these two problems by selecting the keywords simultaneously as a whole group and using a metric (Fisher discriminant or class margin) that is the same as or relevant to the objective function used in the TC classifier.

Another commonly used approach to feature dimension reduction is to learn some linear transforms to project the features onto a low dimensional space. This transform can be learned either supervised by linear discriminative analysis [32] or unsupervised by singular value decomposition (SVD) [33]. Another related approach constructs the low dimensional space more semantically meaningful. For example, the word clustering approach merges semantically related words into word clusters and represents texts in the low dimensional space constructed by the word clusters [34, 35, 36]. The latent semantic indexing (LSI) [37] and its probabilistic variant PLSA [38] follow the same idea but construct the low dimensional space based on some automatically inferred topics. Unfortunately, the transform-based approach is not suitable for the content-aware PTN which is the focus of this work since, in our case, full ASR transcriptions are unavailable.

The last approach to feature dimension reduction is to select prominent features (dimensions) in the process of training the TC classifier [39]. The basic idea is to design a linear classifier where each dimension is assigned a coefficient, and the learning process optimizes the classifier by adjusting the coefficients. The dimensions with sufficiently large coefficients are selected as keywords. This approach solves the problems associated with the intermediate score approach, and is simple and effective [39]. However, cutting off dimensions according to the magnitude of the coefficients, although intuitively reasonable, is not theoretically justified. The sparsity-based approach proposed in this paper follows a similar idea but drives the coefficients of unimportant dimensions to zeros by introducing a sparse constraint, thus avoiding the magnitude-based cutting-off.

2.2 Keyword extraction

This work is also related to document keyword extraction (KE), which is a fundamental task in IR. There is certain overlap between the research of keyword extraction in IR and the dimension reduction in TC, but the former focuses on text representation rather than text discrimination.

The most simple and effective KE approach ranks words using some statistical quantities such as term frequency - inverse document frequency (TFIDF) [40] or distributions of co-occurred words [41]. A large volume of research casts KE to a classification problem (keyword or non-keyword). This approach was first suggested by [42] in the GenEx system and by [43] in the Kea system. Classification models that have been studied include decision trees [44], induction rules [42, 45], naive Bayes [43], conditional random fields (CRFs) [46], maximum entropy models [47], and SVMs [48]. The features that are utilized by the classifier are usually derived from document statistics such as word frequencies and positions [42], but may be also derived from word co-occurrences [41], word coherence [49], linguistic knowledge [45], and semantic knowledge [44, 50].

Recently, graph-based KE approaches have gained popularity, such as the HITS algorithm [51, 23] and the TextRank algorithm [52, 22]. In this approach, a document is represented by a graph where the vertexes represent words and the arcs represent word relationships. The importance of a word is determined by the importance of its neighboring words via a recursive re-estimation algorithm. The word importance, after the re-estimation converging, is used to select the most promising keywords. There are many extensions to the basic graph-based algorithms. [53] combined sentence-level graphs and word-level graphs to infer summaries and keywords simultaneously; [54] extended the TextRank-based approach by considering similar (neighboring) documents; and [55] considered scores calculated based on multiple topics. [23] compared the supervised approach and the graph-based approach, and found that the latter is superior if the training data are limited. A nice review for the recent research on KE can be found in [56].

It should be noticed that most of the KE research so far focuses on text data. Spoken KE research has not been extensively studied. Among the limited exceptions, [57] described a discriminative framework to extract keywords from full transcribed speech signals facilitated by ASR. This is closely related to our research, though we focus on quick detection, and therefore do not rely on full speech transcriptions. [58] followed the same direction but based on the TextRank framework. By introducing a history graph, the authors attained quick adaptation for the keyword group.

3 Sparse analysis

Imposing an l_1 or *lasso* penalty to achieve sparsity on features has been extensively studied in both regression [59, 60] and classification [61, 62, 63, 64]. By adding an

l_1 regularization term to the original cost function, the coefficients of less important feature dimensions are effectively driven to zeros, leading to a natural and efficient feature selection approach which can be used for keyword selection. A remarkable advantage of this sparsity-based approach is that the promising keywords are selected simultaneously as an entire group, which is particularly suitable for the highly restricted keyword selection task of this paper where the group optimization is important.

We investigate two sparse models in this paper: one is based on the simple linear discriminative model while the other is based on the SVM model. The former is simple and efficient, but the latter is more consistent with the TC component, considering that the classifier used in the TC is an SVM in our work.

3.1 Sparse linear discriminative analysis (SDA)

Following the formulation of [61], let $X \in R^{N \times P}$ be a data matrix where N is the number of observations and P is the dimension of the feature vector; further let $Y \in \{0,1\}^{N \times K}$ be the class variables in which Y_{nk} is an indicator variable for which the n-th observation belongs to the k-th class. The optimal scoring criterion for LDA involves recasting the classification problem as a regression problem by turning the categorical target (class label) to a continuous target by multiplying a score vector θ_k. The objective function takes the following form [61]:

$$min_{\beta_k,\theta_k}\{||Y\theta_k - X\beta_k||_2^2\} \quad s.t. \quad \frac{1}{N}\theta_k^T Y^T Y \theta_k = 1, \;\; \theta_k^T Y^T Y \theta_l = 0 \;\; \forall l < k,$$

where θ_k is the K-dimensional score vector, and β_k is a P-dimensional vector of variable coefficients. Note that this is a sequential optimization problem where the 'discriminative directions' $\{\beta_k\}$ are attained one by one. To enforce sparsity in the discriminative directions, [61] appended an l_2 term and an l_1 term to the cost function, given by:

$$\begin{aligned} & min_{\beta_k,\theta_k}\{||Y\theta_k - X\beta_k||_2^2 + \gamma\beta_k^T\Omega\beta_k + \lambda||\beta_k||_1\} \\ & \quad s.t. \quad \frac{1}{N}\theta_k^T Y^T Y \theta_k = 1, \;\; \theta_k^T Y^T Y \theta_l = 0 \;\; \forall l < k, \end{aligned} \tag{1}$$

where Ω is a positive definite matrix to avoid singularity when the observations are mutually dependent or when the dimension is large, i.e., $P > N$, and λ and γ are non-negative hyperparameters. Note that the l_1 penalty introduced by the third term in the above equation enforces sparsity on β_k, and more dimensions of β_k are driven to zeros with a larger λ [59].

In the case of a two-class classification problem such as TC for the content-aware PTN system, there is only one discriminative direction β. The optimization problem is then simplified as follows:

$$\begin{aligned} & min_{\beta,\theta}\{||Y\theta - X\beta||_2^2 + \gamma\beta^T\Omega\beta + \lambda||\beta||_1\} \\ & \quad s.t. \quad \frac{1}{N}\theta^T Y^T Y \theta = 1. \end{aligned} \tag{2}$$

Eliminating θ by a simple calculation leads to:

$$min_{\beta}\{||\hat{Y} - X\beta||_2^2 + \gamma\beta^T\Omega\beta + \lambda||\beta||_1\}, \tag{3}$$

where $\hat{Y}$ is the normalized class indicator matrix whose elements are given by:

$$\hat{Y}_{n,k} = \sqrt{\frac{N}{N_k}},$$

where N_k is the number of observations of the k-th class. We see that the optimization problem for the classification task equals to the optimization problem of a regression task in the case of two classes, which has been stated in [65]. Further, notice that Eq. (3) is an elastic net problem if $\Omega = I$, and a generalized elastic net problem for an arbitrary symmetric positive definite matrix Ω. This elastic net problem can be solved by the algorithm proposed by [60].

Once the optimal β is obtained, for a new observation $x \in R^P$, a simple classification can be conducted by setting a threshold on $\beta^T x$. In this work, however, we treat the SDA as a keyword selector instead of a classifier. First, notice that β is sparse, which indicates that only a fraction of the dimensions of X contributes to the decision. We therefore select the features (words) whose corresponding coefficients in β are not zero as keywords; these keywords are then used to build a new low-dimensional text feature, based on which an SVM (non-linear in this work) is constructed and is used as the classifier for TC.

We finally note that it is only for a binary classification task that the SDA model coincides with the elastic net regression proposed by [60]. For multiple classification tasks, the SDA model is a general framework to derive sparse coefficients $\{\beta_k\}$. In this case, the non-zero dimensions of different β_k are usually different, so the words corresponding to all these non-zero dimensions of all the coefficients $\{\beta_k\}$ have to be selected as keywords.

3.2 Sparse SVM

A shortcoming of the SDA-based keyword selection approach resides in the discrepancy between the objective functions used in the feature selection and the TC classifier: the former is based on the minimum square error, and the latter, which is the SVM in our work, is based on the maximum margin. A better approach is to use the same objective function/model to conduct keyword selection and TC. The sparse SVM model is a good candidate because it is a sparse version of the SVM and both are based on the maximum margin.

We follow the formulation in [65]. First, we define x_n as a training sample and $t_n \in \{+1, -1\}$ as its label. The linear SVM holds a classification boundary $w^T x + b = 0$ where w and b are model parameters, and it predicts the target for x_n (i.e., the category assignment y_n) as follows:

$$y_n = w^T x_n + b. \tag{4}$$

The model training involves optimizing the following regularized hinge function with respect to w and b:

$$C\sum_{n=1}^{N}\xi_n + \frac{1}{2}||w||_2^2 \quad s.t. \quad t_n y_n > 1 - \xi_n, \tag{5}$$

where N is the number of training samples, and ξ_n is a slack variable that represents the cost term of x_n: $\xi_n = 0$ if x_n is inside or on the correct margin boundary, otherwise $\xi_n = |t_n - y_n|$. In addition, $||w||_2^2$ is the regularization term, and C is a tunable hyperparameter to trade off the cost and regularization. From the constraint of Eq. (5), one can show that the distance from the margin to the decision boundary remains to be 1, and so any data x_n is misclassified if $\xi_n > 1$.

As pointed by [63], the l_2 norm $||w||_2^2$ leads to a dense vector of the optimal w. In order to obtain a sparse w, an l_1 norm can be used to substitute for or append to the l_2 norm, leading to the following cost function:

$$\sum_{n=1}^{N}\xi_n + \gamma||w||_2^2 + \lambda||w||_1, \quad s.t. \quad t_n y_n > 1 - \xi_n, \tag{6}$$

where γ and λ are two model hyperparameters for trading off the hinge cost and the regularization. A larger λ drives more dimensions of w to zeros, which in turn vanishes contributions of more features when conducting model inference, according to Eq. (4). Therefore, a sparse SVM leads to a natural way for feature selection. As in SDA, the words corresponding to the non-zero coefficients in w are selected as keywords, and the selected keywords comprise the low dimensional features to build a non-linear SVM model for the TC.

In this work, we employ the template first-order conic solver (TFOCS) to optimize the sparse SVM. TFOCS is a general framework for solving a variety of convex cone problems, including the problem of Eq. (6) [66].

We notice that using a linear sparse SVM to conduct feature selection has been studied in some publications. For example, [63] proposed a quite similar approach to ours, where a linear sparse SVM is used to choose significant dimensions and a non-linear SVM conducts classification. The difference is that [63] worked on v-Support Vector Regression (SVR) and did not involve the l_2 term in Eq. (6). [64] provided another form of sparse SVM, where the maximum number of non-zero dimensions was treated as a constraint, and a convex relaxation approach was employed to optimize the model. To the authors' best knowledge, this paper is the first application of the sparse SVM model to TC keyword selection.

Comparing the SDA-based and sparse SVM-based keyword selection (Eq. (3) and Eq. (6)), we notice that both are based on sparse constraints in the form of an elastic net regularization. The only difference resides in the objective function when optimizing the model coefficients β (in SDA) or w (in sparse SVM): the former is the regularized square error while the latter is the regularized hinge cost. Since the classifier in the TC component is an SVM in this work, the sparse SVM-based approach tends to be more consistent to the TC component.

4 Experimental setup

This section reports the experimental settings and results. As mentioned in the introduction, we make use of a simulation approach to generate the partial ASR output from a text (instead of using a real ASR system), and use an SVM as the classifier in TC. First, we describe the Uyghur text database that is used in the experiments, and then present the text pre-processing steps. The experiments are then presented, which involve a comparative study on three major keyword selection approaches: the intermediate score approach based on TextRank, the intermediate score approach based on document statistics, and the proposed sparsity-based approach.

4.1 Data profile

We choose an Uyghur text database to conduct the experiments. The reason to select Uyghur is that there are real requests to retrieve spoken information in minor languages for the reason of public security, and we hold much interest on keyword selection with limited training data.

Uyghur belongs to the Altai family, Turki branch. Historically, the development of Uyghur can be divided into four stages: the ancient Uyghur, the middle-aged Uyghur, the new-aged Uyghur, and the modern Uyghur. Several symbol systems were adopted in the development process, including Turkic, Uyghur, Arabic, and Chagatay. The modern Uyghur was evolved from the late-period Chagatay, and is a spelling language based on the Arabic alphabet. This is usually called the 'old Uyghur'. In 1970s, the China government tried to create a new writing system for the Uyghur people (sometimes called 'new Uyghur') based on Latin characters but it finally failed to gain popularity. In 1982, the Arabic-based old Uyghur turned into the standard writing system of the modern Uyghur.

The modern Uyghur involves 32 Arabic characters, and each character roughly corresponds to a particular phoneme. Among the 32 phonemes, 8 phonemes are vowels and 24 phonemes are consonants. The writing system involves a number of deformations for each character, depending on the position that the character takes. The writing is word-based and the order is from right to left.

We collected an Uyghur text database that involves 1000 Uyghur documents encoded in the old Uyghur (Arabic characters). These documents were downloaded from several Uyghur websites including ulinix.com and http://www.ts.cn/. 500 documents among them have been labeled as health-related by the web editor, and the rest 500 documents involve multiple topics, including education, history, traffic, environment, economics, computer science, military, and sports. This database has been published online for free download.[1]

We chose 70% of the health & non-health documents (700 in total) as the training data to train the models for keyword selection and the TC classifier, and 10% of the documents (100 in total) as the development set to choose the hyperparameters (e.g., C in SVM); finally, the rest 20% documents (200 in total) were selected as the test data to evaluate the TC performance.

We choose health & non-health as the positive and negative categories in the study. The reason by which we did not choose malicious & non-malicious as the

[1] `http://cslt.riit.tsinghua.edu.cn/mediawiki/index.php/Public_data`

paired categories (which maybe is more desired for a content-aware PTN system) is that the definition of a ‘malicious conversation’, to our understanding, is rather subjective and so it is not quite suitable for scientific research. Nevertheless, we believe that the conclusions obtained here can be well generalized to other tasks including malicious content detection.

4.2 Uyghur text pre-processing

The text pre-processing for Uyghur documents involves three steps: character purge, Latinization, and stop words removal.

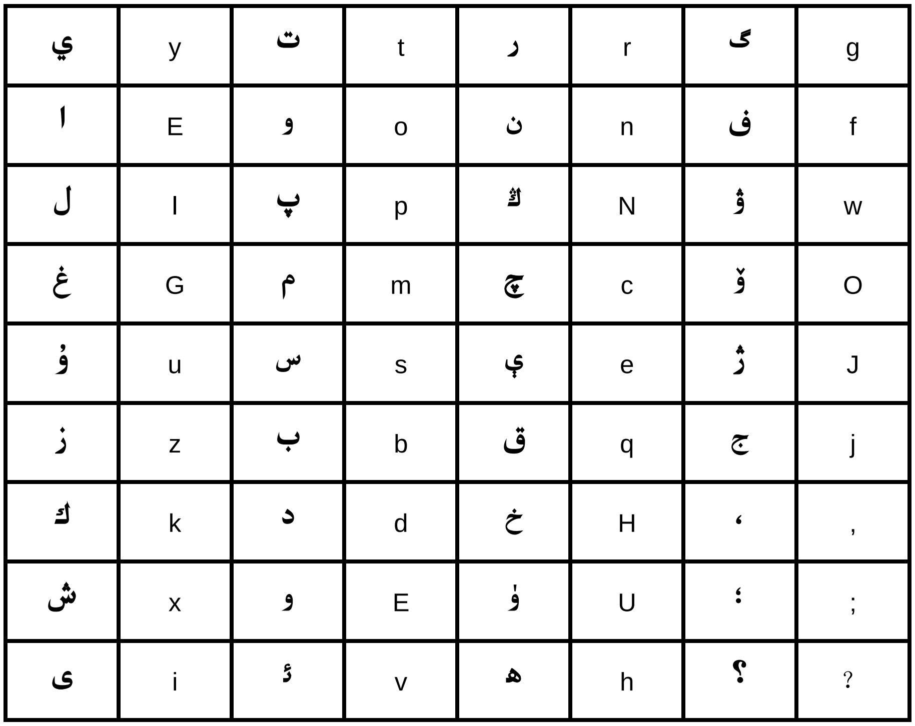
- Character purge
  The first step removes less informative tokens, including punctuations, digits, mathematic symbols, and special symbols such as ‘$’ and ‘#’. These characters involve little semantic information and introduce much noise if involved in the feature vector, so they need to be removed.
  As already mentioned, the ASR output is generated by a simulation method instead of by a real ASR system. This is achieved by a special step for all the *evaluation* documents: converting all the non-keywords to the trivial filler, i.e., the ‘np’ symbol.
- Latinization
  The Uyghur documents we collected are in old Uyghur, which are based on Arabic characters and so are not suitable for computer-based processing. In order to simplify the following processing, we convert the Arabic characters of the old Uyghur to Latin characters. A mapping rule designed by the Intelligent Information Processing Lab (IIPL) of Xinjiang University was used to perform the conversion, as shown in Fig. 2.
- Stop word removal
  Stop words can be categorized into two classes. The first class involves word-s that represent little semantic information, such as conjunctions, pronouns, quantifications, and interjections. The second class involves words that are equally distributed in all texts and so are little discriminative for text categories. The stop words that we removed are listed in Fig. 3, where we use the old Uyghur for easy reading.

4.3 TextRank-based keyword selection

The first experiment studies the TetxRank-based keyword selection method. Firstly, all the health documents are merged into a single document, and then the TextRank algorithm is performed to compute the centrality of every word. Note that only the health documents are used here because our goal is to extract the words that are representative for the health category. The words with top-n centralities are then selected as n keywords. The TextRank package implemented in Perl 5 was used to conduct the computation[2], where the connection degrees between words are initialized by word co-occurrences.

Once the keywords are chosen, the keywords can be used to extract features and build the TC classifier. In our experiments, we found that the simple term frequency (TF) feature performs slightly better than the commonly used TFIDF

[2] http://search.cpan.org/dist/Text-Categorize-Textrank/lib/Text/Categorize/Textrank/En.pm

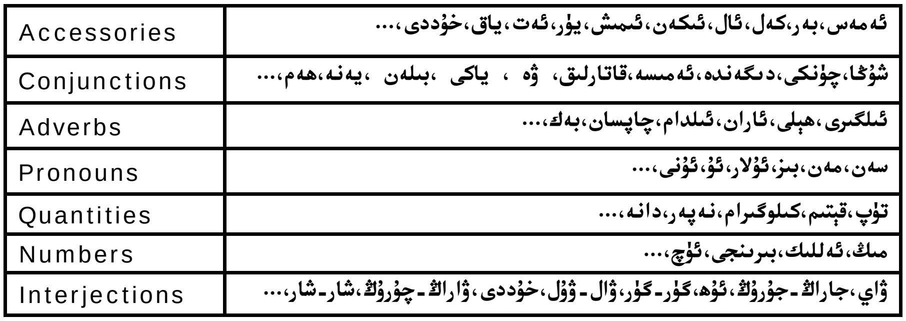


ي	y	ت	t	ر	r	گ	g
ا	E	و	o	ن	n	ف	f
ل	l	پ	p	ڭ	N	ۋ	w
غ	G	م	m	چ	c	ۆ	O
ۇ	u	س	s	ې	e	ژ	J
ز	z	ب	b	ق	q	ج	j
ك	k	د	d	خ	H	،	,
ش	x	و	E	ۈ	U	؛	;
ى	i	ئ	v	ھ	h	؟	?

Figure 2 Character mapping from old Uyghur to Latin characters.

feature, which is perhaps due to the unreliable document frequencies computed with the limited training data (700 documents). We therefore used the TF feature in all the following experiments of the paper. Note that here we mean the feature used for the TC classifier; for keyword selection, a multitude of features can be used, as we will present shortly.

As mentioned already, we choose the SVM model as the TC classifier. The Gaussian kernel is used in our work and the model hyperparameter C is optimized with the development set. Once the SVM has been trained, it is used to conduct TC on the evaluation data, for which the classification accuracy (CA) is used to evaluate



Accessories	ئەمەس،بەر،كەل،ئال،ئىكەن،ئىمىش،يۈر،ئەت،ياق،خۇددى،...
Conjunctions	شۇڭا،چۈنكى،دىگەندە،ئەمسە،قاتارلىق، ۋە ، ياكى ،بىلەن ،يەنە،ھەم،...
Adverbs	ئىلگىرى،ھېلى،ئاران،ئىلدام،چاپسان،بەك،...
Pronouns	سەن،مەن،بىز،ئۇلار،ئۇ،ئۇنى،...
Quantities	تۈپ،قېتىم،كىلوگىرام،نەپەر،دانە،...
Numbers	مىڭ،ئەللىك،بىرىنچى،ئۈچ،...
Interjections	ۋاي،جاراڭ-جۇرۇڭ،ئۇھ،گۈر-گۈر،ۋال-ۋۇل،خۇددى،ۋاراڭ-چۇرۇڭ،شار-شار،...

Figure 3 Stop words in Uyghur.

the TC performance. In our work, the libSVM tool[3] was used to conduct the model training and perform classification.

Fig. 4 presents the CA results with the TextRank-based keyword selection method, where the number of keywords n varies from 20 to 100. We observe that the CAs are between 70% and 75%, which is a rather low performance for a binary classification task. This can be attributed to the fact that the TextRank-based selection ignores the non-health documents and considers only the health documents, which probably results in keywords that are more representative than discriminative. This in turn leads to a suboptimal TC model. Additionally, we observe that the CA curve is rather 'bumpy', suggesting that the keywords selected are not group-optimal: a new selected keyword may reduce the performance of the group.

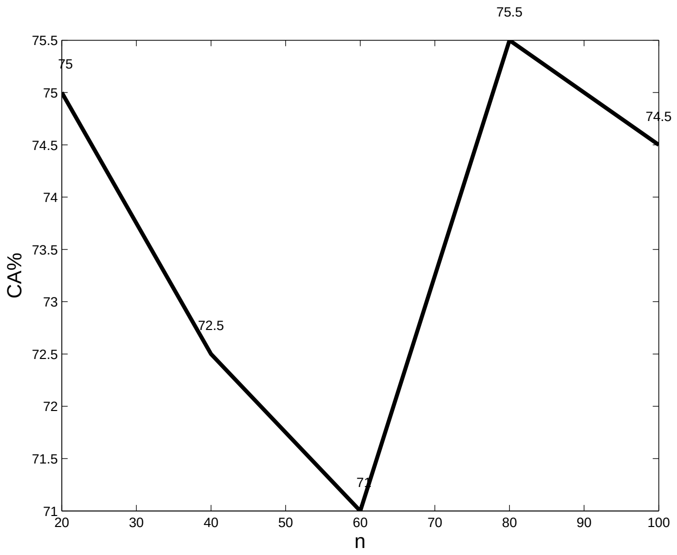


Figure 4 CA results with TextRank-based keyword selection. The TF is used as the document feature, and the SVM is used as the classification model.

4.4 Keyword selection based on document statistics

In the second set of experiments, we study the keyword selection based on document statistics. [18] found that word selection based on these statistics may lead to highly competitive TC performance when compared with some deliberately designed intermediate scores such as IG and MI, but the former is much simpler.

4.4.1 Non-discriminative statistics

We first experiment with non-discriminative statistics, e.g., those statistics that represent properties of the health documents. They are, therefore, similar to the TextRank-based scores in this sense. Four statistics are experimented: DF on health documents (DF_h), TF on health documents (TF_h), $DF_h * TF_h$, and $DF_h * TF_h * IDF_{h+n}$ where IDF_{h+n} is the inverse document frequency (IDF) value computed on both the health and non-health documents. Most of the statistics are

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

straightforward; the only one that requires a bit explanation is the fourth, where IDF_{h+n} is involved in order to reduce the impact of frequent words in both categories. Note that although the non-health documents are involved here, the statistics are still representative for the health category.

	CA%				
n	20	40	60	80	100
DF_h	88.0	87.5	88.5	89.5	89.0
TF_h	90.5	92.5	89.5	89.5	90.0
$TF_h * DF_h$	91.0	91.5	92.5	90.0	93.0
$TF_h * DF_h * IDF_{h+n}$	93.0	92.5	90.0	90.5	92.5

Table 1 CA results with keyword selection based on non-discriminative document statistics. 'n' refers to the number of keywords.

These statistics are computed for each word in the training data, based on which the keywords are selected. The TC model (SVM) is then trained with the TF feature, and is evaluated as in the TextRank-based experiment. Table 1 presents the CA results. The results show that both DF_h and TF_h can be used as the intermediate score to select keywords, but their combination is more effective. Applying the global IDF_{h+n} seems to improve performance in some circumstances, but it might lead to performance reduction in others. When compared with the TextRank-based keyword selection approach, we find that the document statistics-based approach is much more effective, suggesting that the TextRank is perhaps not suitable for TC, at least in some circumstances.

4.4.2 Discriminative statistics

In order to improve the discriminative capability of the selected keywords, we design a number of 'discriminative statistics'. These statistics take into account both the health and non-health documents, and so is more related to the TC objective.

Table 2 presents the discriminative statistics and the corresponding TC performance. Note that the subscript n indicates that the statistics are computed with the non-health documents. It can be observed that the discriminative statistics generally perform better than the non-discriminative statistics as shown in Table 1, confirming that discriminative information, even in a very simple form, can lead to better keyword selection.

	CA%				
n	20	40	60	80	100
$DF_h - DF_n$	91.0	93.5	91.5	93.5	92.0
$TF_h - TF_n$	93.5	92.5	90.0	92.5	92.0
$TF_h * DF_h - TF_n * DF_n$	91.5	91.0	95.0	94.0	91.5
$TF_h * DF_h * IDF_{h+n} - TF_n * DF_n * IDF_{h+n}$	90.5	90.5	89.5	91.0	94.0

Table 2 CA results with keyword selection based on discriminative document statistics. 'n' denotes the number of keywords.

Although simple and promising, the document statistics-based approach shares the same problem with the TextRank-based approach: the CA results are highly 'bumpy' with n increasing. This means that the keyword selection is rather unreliable, and a good CA obtained with a particular keyword group may be simply lost if another keyword is added into the group. This uncertainly is a common problem of all the keyword selection approaches based on intermediate scores.

4.5 Sparse discriminative keyword selection

The last experiment studies two sparse models: SDA and sparse SVM. As discussed in Section 3, these two models share the same idea to build an l_1 regularized linear model with which prominent keywords can be selected simultaneously according to the model coefficients. This differs from the two approaches used in the previous sections where the keywords are selected independently. The difference between the SDA and the sparse SVM is that the objective function of the SDA training is the least square error, while the sparse SVM training targets to maximize the class margin.

Similar to the approach based on discriminative statistics, both the health and non-health documents are employed in the sparsity-based approach, so both the two approaches are discriminative in nature. A significant difference is that, the sparsity-based approach utilizes the class discriminative information to train a discriminative model, while the discriminative statistics-based approach designs some intermediate scores.

In theory, any feature can be used to train the sparse model since the keyword selection and the TC are two separate components; however, choosing the same feature for the two components may lead to better consistency. We therefore choose the TF feature when training the SDA and the sparse SVM models.

In this work, the SDA model was trained with a tool provided by Line Clemmensen[4]. The tool was developed in MATLAB and depends on the LARSEN algorithm [60] implemented in the SpaSM toolbox[5]. In this experiment, γ in Eq. (3) is optimized with the development set, and λ is selected so that the required number of keywords is exactly extracted. The CA results with the SDA-based keyword selection method are presented in Fig. 5.

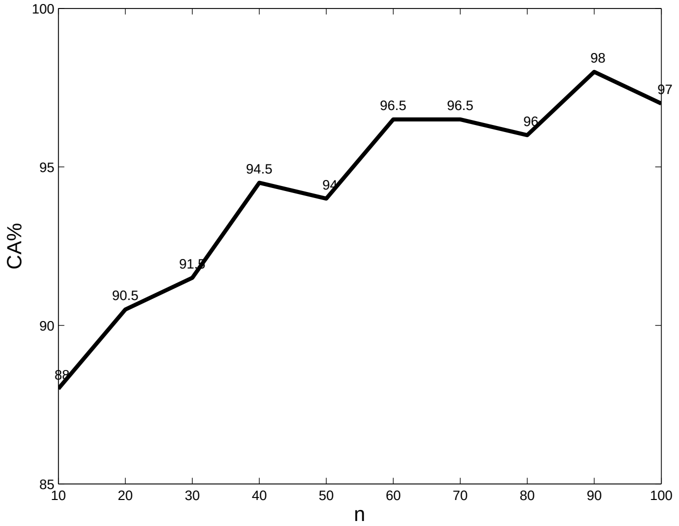


Figure 5 CA results with SDA-based keyword selection. ‘n’ stands for the number of keywords.

[4] `http://www.imm.dtu.dk/~lkhc/indexP.html`

[5] http://www2.imm.dtu.dk/projects/spasm/

For the sparse SVM, we utilized the implementation in the TFOCS toolbox[6]. γ and λ are chosen in the same way as in SDA. The CA results with the sparse SVM-based keyword selection method are presented in Fig. 6.

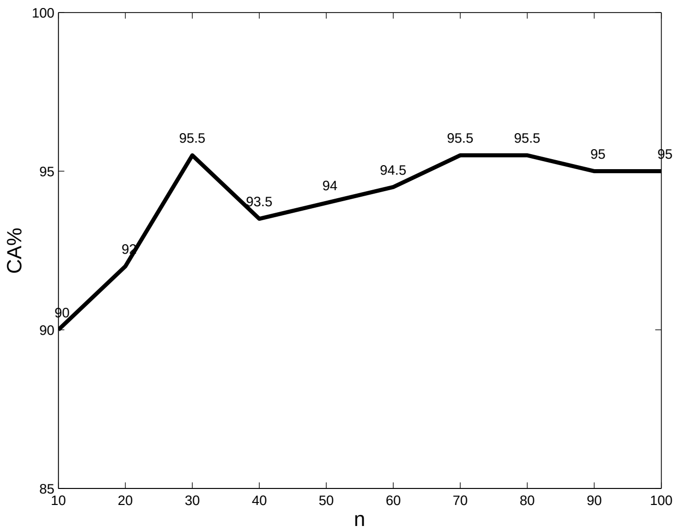


Figure 6 CA results with sparse SVM-based keyword selection. 'n' stands for the number of keywords.

From the results obtained with the SDA and the sparse SVM, we observe that both of the two sparse models provide better performance than the TextRank and document statistics-based approaches. More importantly, the CA results are more consistent: with more keywords selected, the TC performance is generally improved. This solves the problem associated with the intermediate score-based approaches and provides a reliable keyword selection method.

Comparing the SDA and the sparse SVM, we see that they perform similar, though the CA curve with the sparse SVM seems 'smoother' than with the SDA. Particularly, the sparse SVM seems to be superior to the SDA when the number of keywords is small. This supports our argument that the sparse SVM is theoretically more consistent with the SVM-based TC.

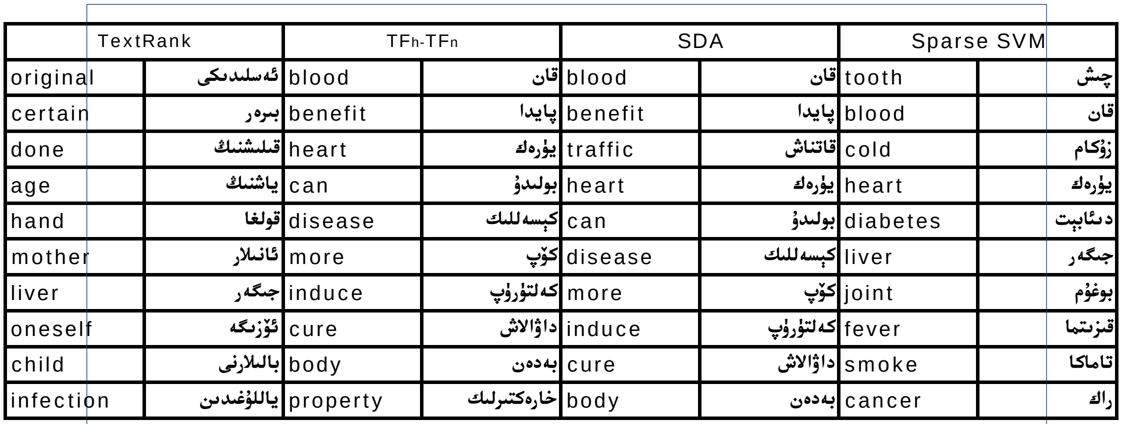
4.6 Keyword comparison

To have a more intuitive comparison, the top 10 keywords selected with each of the four selection methods are given in Fig. 7. Note that the column '$TF_h - TF_n$' represents the approach based on discriminative document statistics. It can be clearly seen that the sparsity-based keyword selection, particularly the sparse SVM-based approach, provides more informative keywords than the other approaches.

5 Conclusions

This paper presented a highly restricted keyword selection approach based on sparse analysis for text categorization. The goal is to deliver a reliable keyword selection technique for applications which rely on a few but informative keywords such as the

[6] `http://cvxr.com/tfocs/`



TextRank		TF_h-TF_n		SDA		Sparse SVM	
original	ئەسلىدىكى	blood	قان	blood	قان	tooth	چىش
certain	بىرەر	benefit	پايدا	benefit	پايدا	blood	قان
done	قىلىشنىڭ	heart	يۈرەك	traffic	قاتناش	cold	زۇكام
age	ياشنىڭ	can	بولىدۇ	heart	يۈرەك	heart	يۈرەك
hand	قولغا	disease	كېسەللىك	can	بولىدۇ	diabetes	دىئابېت
mother	ئانىلار	more	كۆپ	disease	كېسەللىك	liver	جىگەر
liver	جىگەر	induce	كەلتۈرۈپ	more	كۆپ	joint	بوغۇم
oneself	ئۆزىگە	cure	داۋالاش	induce	كەلتۈرۈپ	fever	قىزىتما
child	بالىلارنى	body	بەدەن	cure	داۋالاش	smoke	تاماكا
infection	ياللۇغدىن	property	خارەكتېرلىك	body	بەدەن	cancer	راك

Figure 7 Top-10 keywords selected with different keyword selection methods.

content-aware PTN. We proposed to use a linear sparse model to conduct keyword selection, and argued that this sparsity-based approach leads to an elegant way to select keywords in a group-optimized way.

We verified the proposal with an Uyghur text database, and experimented with two sparse models: SDA and sparse SVM. The experimental results demonstrated that both the SDA and the sparse SVM lead to better TC performance than the conventional approaches based on intermediate scores (for which we experimented with the TextRank score and document statistics). More importantly, the sparse approach produces more consistent and more reliable TC results than the conventional approaches when the number of keywords varies, confirming our conjecture that the sparse approach selects keywords in a group-optimal way.

Although the focus of this work is keyword selection, the sparse approach is a general tool for selecting prominent features. The future work will study sparsity-based heterogeneous feature selection for TC. Another work in the near future is to test the proposed approach with a real ASR system, for which false detections and missings within the ASR output have to be considered.

Acknowledgement

This research was supported by the National Science Foundation of China (NSFC) under the projects No. 61371136 and No. 61163033. It was also partially supported by the Program for New Century Excellent Talents in Universities of China (NCET-10-0969). Part of the work was supported by Sinovoice and Huilan Ltd.

Author details

[1]Center for Speech and Language Technology, Research Institute of Information Technology, Tsinghua University, ROOM 1-303, BLDG FIT, 100084 Beijing, China. [2]Center for Speech and Language Technologies, Division of Technical Innovation and Development, Tsinghua National Laboratory for Information Science and Technology, ROOM 1-303, BLDG FIT, 100084 Beijing, China. [3]Department of Computer Science and Technology, Tsinghua University, ROOM 1-303, BLDG FIT, 100084 Beijing, China. [4]GEINTRA, University of Alcalá, Madrid, Spain.

References

1. Mark Gales and Steve Young, "The application of hidden markov models in speech recognition," *Foundations and Trends in Signal Processing*, vol. 1, no. 3, pp. 195–304, 2007.
2. Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.
3. Vladimir Vapnik, *The nature of statistical learning theory*, Springer, New York, 1995.
4. Thorsten Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Proceedings of the European Conference on Machine Learning*, 1998, pp. 136–142.
5. Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen, *Classification and regression trees*, Chapmon and Hall/CRC, 1984.
6. Shrikanth Shankar and George Karypis, "A feature weight adjustment algorithm for document categorization," in *Proceedings of the International Workshop on Multimedia Data Mining*, 2000.
7. Wenqian Shang, Houkuan Huang, Haibin Zhu, Yongmin Lin, Youli Qu, and Zhihai Wang, "A novel feature selection algorithm for text categorization," *Expert Systems with Applications*, vol. 33, no. 1, pp. 1–5, 2007.
8. J. Ross Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
9. Tom M Mitchell, *Machine learning*, McGraw-Hill, 1997.
10. David D Lewis, "Feature selection and feature extraction for text categorization," in *Proceedings of the workshop on Speech and Natural Language*, 1991, pp. 212–217.
11. Susan Dumais, John Platt, David Heckerman, and Mehran Sahami, "Inductive learning algorithms and representations for text categorization," in *Proceedings of the international conference on Information and knowledge management*, 1998, pp. 148–155.
12. Ted Dunning, "Accurate methods for the statistics of surprise and coincidence," *Computational Linguistics*, vol. 19, no. 1, pp. 61–74, 1993.
13. Jingnian Chen, Houkuan Huang, Shengfeng Tian, and Youli Qu, "Feature selection for text classification with naïve bayes," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5432–5435, 2009.
14. Igor Kononenko, "On biases in estimating multi-valued attributes," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 1995, pp. 1034–1040.
15. Dunja Mladenić and Marko Grobelnik, "Feature selection on hierarchy of web documents," *Decision Support Systems*, vol. 35, no. 1, pp. 45–87, 2003.
16. C. J. Van Rijsbergen, David J. Harper, and Martin F. Porter, "The selection of good search terms," *Information Processing & Management*, vol. 17, no. 2, pp. 77–91, 1981.
17. Daphne Koller and Mehran Sahami, "Hierarchically classifying documents using very few words," in *Proceedings of the International Conference on Machine Learning*, 1997, pp. 170–178.
18. Yiming Yang and Jan O Pedersen, "A comparative study on feature selection in text categorization," in *Proceedings of the International Conference on Machine Learning*, 1997, pp. 412–420.
19. W John Wilbur and Karl Sirotkin, "The automatic identification of stop words," *Journal of Information Science*, vol. 18, no. 1, pp. 45–55, 1992.
20. Yiming Yang, "Noise reduction in a statistical approach to text categorization," in *Proceedings of the annual international ACM SIGIR conference on Research and development in information retrieval*, 1995, pp. 256–263.
21. Toru Hisamitsu and Yoshiki Niwa, "A measure of term representativeness based on the number of co-occurring salient words," in *Proceedings of the International Conference on Computational Linguistics*, 2002, pp. 1–7.
22. Rada Mihalcea and Paul Tarau, "Textrank: Bringing order into texts," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2004, pp. 404–411.
23. Marina Litvak and Mark Last, "Graph-based keyword extraction for single-document summarization," in *Proceedings of the workshop on Multi-source Multilingual Information Extraction and Summarization*, 2008, pp. 17–24.
24. Wei Zhao, Yafei Wang, and Dan Li, "A new feature selection algorithm in text categorization," in *Proceedings of International Symposium on Computer Communication Control and Automation*, 2010, pp. 146–149.
25. Fabrizio Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.
26. Baharum Baharudin, Lam Hong Lee, and Khairullah Khan, "A review of machine learning algorithms for text-documents classification," *Journal of advances in information technology*, vol. 1, no. 1, pp. 4–20, 2010.
27. CharuC. Aggarwal and ChengXiang Zhai, "A survey of text classification algorithms," in *Mining Text Data*, Charu C. Aggarwal and ChengXiang Zhai, Eds., pp. 163–222. Springer US, 2012.

28. Vandana Korde and C Namrata Mahender, "Text classification and classifiers: A survey," *International Journal of Artificial Intelligence & Application*, vol. 3, no. 2, pp. 85–99, 2012.
29. Ali Danesh, Behzad Moshiri, and Omid Fatemi, "Improve text classification accuracy based on classifier fusion methods," in *Proceedings of the International Conference on Information Fusion*, 2007, pp. 1–6.
30. David E. Johnson, Frank J. Oles, Tong Zhang, and Thilo Goetz, "A decision-tree-based symbolic rule induction system for text categorization," *IBM Systems Journal*, vol. 41, no. 3, pp. 428–437, 2002.
31. Cheng Hua Li and Soon Choel Park, "An efficient document classification model using an improved back propagation neural network and singular value decomposition," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3208–3215, 2009.
32. Soumen Chakrabarti, Shourya Roy, and Mahesh V Soundalgekar, "Fast and accurate text classification via multiple linear discriminant projections," *The International Journal on Very Large Data Bases*, vol. 12, no. 2, pp. 170–185, 2003.
33. Peg Howland and Haesun Park, "Generalizing discriminant analysis using the generalized singular value decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 995–1006, 2004.
34. L Douglas Baker and Andrew Kachites McCallum, "Distributional clustering of words for text classification," in *Proceedings of the annual international ACM SIGIR conference on Research and development in information retrieval*, 1998, pp. 96–103.
35. Noam Slonim and Naftali Tishby, "The power of word clusters for text classification," in *Proceedings of the European Colloquium on Information Retrieval Research*, 2001.
36. Ron Bekkerman, Ran El-Yaniv, Naftali Tishby, and Yoad Winter, "On feature distributional clustering for text categorization," in *Proceedings of the annual international ACM SIGIR conference on Research and development in information retrieval*, 2001, pp. 146–153.
37. Scott Deerwester, Susan T Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.
38. Thomas Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the annual international ACM SIGIR conference on Research and development in information retrieval*, 1999, pp. 50–57.
39. Dunja Mladenić, Janez Brank, Marko Grobelnik, and Natasa Milic-Frayling, "Feature selection using linear classifier weights: interaction with classification models," in *Proceedings of the annual international ACM SIGIR conference on Research and development in information retrieval*, 2004, pp. 234–241.
40. Gerard Salton and Christopher Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.
41. Yutaka Matsuo and Mitsuru Ishizuka, "Keyword extraction from a single document using word co-occurrence statistical information," *International Journal on Artificial Intelligence Tools*, vol. 13, no. 01, pp. 157–169, 2004.
42. Peter Turney, "Learning to extract keyphrases from text," *Technical Report, Institute for Information Technology, National Research Council*, 1999.
43. Eibe Frank, Gordon W Paynter, Ian H Witten, Carl Gutwin, and Craig G Nevill-Manning, "Domain-specific keyphrase extraction," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 1999, pp. 668–673.
44. Gonenc Ercan and Ilyas Cicekli, "Using lexical chains for keyword extraction," *Information Processing & Management*, vol. 43, no. 6, pp. 1705–1714, 2007.
45. Anette Hulth, "Improved automatic keyword extraction given more linguistic knowledge," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2003, pp. 216–223.
46. Chengzhi Zhang, "Automatic keyword extraction from documents using conditional random fields," *Journal of Computational Information Systems*, vol. 4, no. 3, pp. 1169–1180, 2008.
47. Su Nam Kim and Min-Yen Kan, "Re-examining automatic keyphrase extraction approaches in scientific articles," in *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, 2009, pp. 9–16.
48. Mikalai Krapivin, M Autayeu, Maurizio Marchese, Enrico Blanzieri, and Nicola Segata, "Improving machine learning approaches for keyphrases extraction from scientific documents with natural language knowledge," in *Proceedings of the International Conference on Asia-Pacific Digital Libraries*, 2010, pp. 102–111.
49. Peter Turney, "Coherent keyphrase extraction via web mining," in *Proceedings of the International Joint Conference on Artificial intelligence*, 2003, pp. 434–439.
50. Yun-Nung Chen, Yu Huang, Sheng-Yi Kong, and Lin-Shan Lee, "Automatic key term extraction from spoken course lectures using branching entropy and prosodic/semantic features," in *Proceedings of the IEEE Spoken Language Technology Workshop (SLT)*, 2010, pp. 265–270.
51. Jon M Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604–632, 1999.
52. Sergey Brin and Lawrence Page, "The anatomy of a large-scale hypertextual web search engine," *Computer networks and ISDN systems*, vol. 30, no. 7, pp. 107–117, 1998.
53. Xiaojun Wan, Jianwu Yang, and Jianguo Xiao, "Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2007, pp. 552–559.
54. Xiaojun Wan and Jianguo Xiao, "Exploiting neighborhood knowledge for single document summarization and keyphrase extraction," *ACM Transactions on Information Systems (TOIS)*, vol. 28, no. 2, pp. 8:1–8:34, 2010.
55. Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun, "Automatic keyphrase extraction via topic decomposition," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2010, pp. 366–376.
56. Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin, "Automatic keyphrase extraction from scientific articles," *Language resources and evaluation*, vol. 47, no. 3, pp. 723–742, 2013.
57. Fei Liu, Feifan Liu, and Yang Liu, "A supervised framework for keyword extraction from meeting transcripts,"

IEEE Transactions on Audio, Speech, and Language Processing, vol. 19, no. 3, pp. 538–548, 2011.

58. Hyun-Je Song, Jun-Ho Go, Seong-Bae Park, and Se-Young Park, "A just-in-time keyword extraction from meeting transcripts," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2013, pp. 888–896.
59. Robert Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B*, pp. 267–288, 1996.
60. Hui Zou and Trevor Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B*, vol. 67, no. 2, pp. 301–320, 2005.
61. Line Clemmensen, Trevor Hastie, Daniela Witten, and Bjarne Ersbøll, "Sparse discriminant analysis," *Technometrics*, vol. 53, no. 4, pp. 406–413, 2011.
62. Daniela M Witten and Robert Tibshirani, "Penalized classification using fisher's linear discriminant," *Journal of the Royal Statistical Society: Series B*, vol. 73, no. 5, pp. 753–772, 2011.
63. Jinbo Bi, Kristin P. Bennett, Mark Embrechts, Curt Breneman, and Minghu Song, "Dimensionality reduction via sparse support vector machines," *The Journal of Machine Learning Research*, vol. 3, pp. 1229–1243, 2003.
64. Mingkui Tan, Li Wang, and Ivor W Tsang, "Learning sparse SVM for feature selection on very high dimensional datasets," in *Proceedings of the International Conference on Machine Learning*, 2010, pp. 1047–1054.
65. Christopher M Bishop, *Pattern recognition and machine learning*, vol. 1, Springer, New York, 2006.
66. Stephen R Becker, Emmanuel J Candès, and Michael C Grant, "Templates for convex cone problems with applications to sparse signal recovery," *Mathematical Programming Computation*, vol. 3, no. 3, pp. 165–218, 2011.