Asian Journal of Information Technology 5 (4) : 402-407, 2006

Automatic Keyphrase Extraction Using Probabilistic Prediction

S.M. Rafizul Haque, Khalid Al Mustansir Billah and Md. Mahamudul Haque
Computer Science and Engineering Discipline,
School of Science, Engineering and Technology, Khulna University, Khulna, Bangladesh

Abstract: This study proposes an approach to Automatic Keyphrase Extraction depending on the semantic coherence between the phrases in a document. The approach uses a probabilistic index added to existing measures to predict the occurrence of keyphrases before the processing. The approach is suitable for both single and multiple word phrases and uses the measures of semantic coherence and baseline metrics to calculate the relative weights for the phrases as well as using a probabilistic measure introduced to improve performance. This probabilistic measure is intended to reduce the complexity of identifying a keyphrase considering its likeliness to occur in an index position.

Key words: Keyphrase extraction, semantic coherence, probabilistic index

INTRODUCTION

A keyphrase is the phrase of particular importance in a document that contains close relation with the document's contents and context. Keyphrases can be one word long, or may comprise of more than one word. Keyphrases can serve as a highly condensed summary, they can supplement or replace the title as a label for the document, and, in addition, they can be highlighted in the body of the text, to speed up reading[1].

Keyphrases serve various purposes from providing a brief summary of the content to measuring relevance and context of the document. In most cases, a technical paper is associated with a number of keyphrases to make the intention and content clear. For a collection of documents, keyphrases can be used for indexing, categorizing, clustering, browsing, or searching[1].

The term Automatic Keyphrase Extraction refers to the selection of keyphrases from a document by means of an automated system or an algorithm. These systems often need some training to be able to process the documents in one field. The system is trained by the processing of some manually tagged keyphrases in a document and then is used to process the other documents.

The vast amount of existing documents in various fields often does not have keyphrases associated with them. It becomes a quite tedious job to manually associate the keyphrases needed for them. So, in the field of Information Extraction and Data Mining, the development of new strategies for automatic keyphrase extraction has always been of great interest. In this study, a proposal is made to use a probabilistic prediction to determine the relevance of a keyphrase in a document.

RELATED WORKS

The field of keyphrase extraction, being one of the rapid developing areas of computation, has experienced many works of different classes to do the job. Frank *et al.* and Turney used keyphrase extraction methodologies as a supervised learning problem, Tomokiyo and Hurst[2] used the concept of phraseness and informativeness and a language model approach to extract domain specific keyphrases and Dunning described the Binomial Log likelihood Ratio Test as a measure to collocation discovery. Witten *et al.*[3], on the other hand, developed the KEA algorithm to extract keyphrases.

All of the algorithms have their advantages and limitations. This paper proposes an approach with a combined approach of statistical measures with supervised learning. The statistical measures are taken from the proposal of Tomokiyo and Hurst about the consideration of the relevance of a keyword using the combination of phraseness and informativeness scores and phrase ranking.

In previous works, the keyphrases are extracted based on the statistical measures[2] or on the knowledge obtained from the training phase. In statistical approach, it is much easier to extract keyphrases, as previous knowledge does not need to be arranged.

Corresponding Author: Rafizul Haque, S.M., Computer Science and Engineering Discipline, School of Science, Engineering and Technology, Khulna University, Khulna, Bangladesh

USE OF PROBABILISTIC MEASURES TO EXTRACT KEYPHRASES

This study suggests a combination of the statistical assumption techniques with supervised learning and proposes a probability-based method to implement the supervised learning technique. The result of the combination is faster extraction of keyphrases with greater accuracies.

The new proposal presented in this paper has two distinct parts. First, it proposes the combination of statistical techniques with supervised learning to make better use of the properties of both approaches while suppressing their limitations. And second, it proposes a probability-based method to implement supervised learning.

The new approach might seem similar to an N-gram language model. But it is different in the sense that, in an N-gram model, the probability of a word being a keyphrase depends on the sequence of previous N words. Where, the proposed approach, a word's relevance is measured after it is encountered based on some probabilistic value calculated from the previously found keyphrases.

Structure of the keyphrases: To make it suitable for the predictive approach, keyphrases are considered as a sequence of keywords with optional prepositions in between them. Thinking keyphrases as a sequence of words make it possible for the association graph to manipulate them.

Where, K_i is a keyword, P_i is either a empty or a non-significant word.

The association graph: The proposed approach utilizes the measures for prediction of phrases coming after certain phrase. The initial phrase can be repeated in the later segment of text as well (such as 'as' in the phrase 'as well as'). This situation can easily be modeled using a graph. The graph should contain nodes for the keyphrases and the weighted edges representing the probability.

The graph includes nodes for each phrase and the edges from a phrase K_i to another K_j if K_j is relevant to come after K_i. The weight of the edge indicates the probability to which K_j is likely to come after K_i.

$$K_p = P_0K_1P_1K_2P_2\ldots\ldots K_n$$

Fig 1: Structure of the keyphrases

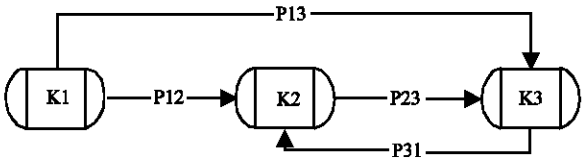


Fig. 2: The association graph

In the association graph of Fig. 2, each probability P_{ij} is the probability for Keyphrase K_j to occur after Keyphrase K_i

In this approach, when any keyphrase is encountered, a set containing the next most likely keyphrases is made and the next keyphrase is compared against that set. If it is contained in the set, then it must have been predicted by the previous results and therefore, it can be marked as a keyphrase. On the other hand, if it is not predicted by the set, then its validity needs to be determined by the other methods.

The system: The system may be considered of having two different stages, namely, the training stage and the processing stage. In the training stage, the system is prepared for processing by the build up of the association graph, modification of the knowledge base, registration of keywords and by making the system 'familiar' with the certain domain for processing. The training mechanism enables the system to work in completely different domains of interest if it is trained in that domain prior to processing.

From Fig. 1, it is apparent that, the system prepares itself for the processing phase in the training phase using some training documents and does the real processing after the preparation. Thresholds and limits provide the values for the selection of keyphrases. The threshold

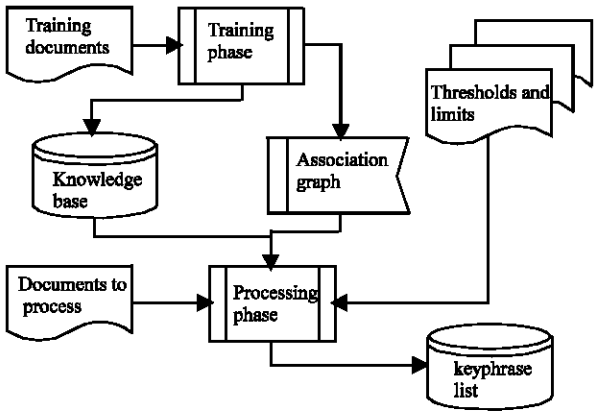


Fig. 3: The system

values can be used to tune the sensitivity of the system to certain degrees. The term threshold is used to mean the differentiating mark among the values of acceptable range to indicate level of performance. And the term Limit is used to differentiate between the regions of acceptable and non-acceptable values.

Both phases comprise of a number of sub-phases. For the training phase, different sub-phases complete different tasks of Data Cleaning, Building a list for support and confidence, construction of FP tree, determining the phrase from the FP-tree and build up of the association graph. The processing phase, on the other hand, has the tasks of data cleaning, determination and identification, extraction of keyphrases and representation of result.

Training the system: The algorithm, like many others in this field, needs to be trained before the processing by some documents with manually pointed out keyphrases. The system builds up the graph in this training session. Various edges of the graph are created and labeled in the process as each of the keyphrases is identified. The complete graph then resembles the relation between the keyphrases in an orderly fashion.

For example, if any term y is to be occurred after x, we have an edge in the graph from x to y and as the system goes on processing, each occurrence of x and y counts in the probability of occurrence of y after x and thus the edge weight changes.

The principal difference of this method with most other methods is, the training procedure is automated as well as the processing. In most approaches, the training phase requires some documents with manually tagged keyphrases to make the system learn the relevance of a candidate phrase to become a keyphrase.

Selection factors: The term phraseness means a degree to which a given word sequence is considered to be a phrase and informativeness refers to how well a phrase illustrates the key ideas in a document[2]. A combination of phraseness and informativeness of a word can be a measure to its validity as a keyphrase.

Eq. (1), (2) and (3) are adopted from the paper of Tomokiyo and Hurst[1] for measuring the informativeness and phraseness of a word using Binomial Log likelihood Ratio Test (BLRT).

Let, in a corpus, one word is observed k_1 times out of n_1 tokens as drawn from different distributions and k_2 times out of n_2 tokens as drawn from same distribution. Then, the likelihood ratio of the word, R will be

$$R = 2\log\frac{L(p_1,k_1,n_1)L(p_2,k_2,n_2)}{L(p,k_1,n_1)L(p,k_2,n_2)} \qquad (1)$$

where $p_i = \frac{k_i}{n_i}$ and $p = \frac{k_1 + k_2}{n_1 + n_2}$

and

$$L(p,k,n) = p^k(1-p)^{n-K} \qquad (2)$$

As described in[2], for calculating phraseness for any pair of words x, y we have to set

$$k_1 = C(x, y)$$
$$n_1 = C(x)$$
$$k_2 = C(\sim x, y) = C(y) - C(x, y)$$
$$n_2 = C(\sim x) = \sum_w C(w) - C(x)$$

Where,

C(x) = Frequency of word x,
C (x, y) = frequency of y following x

For informativeness, we have,

$$k_1 = C_{fg}(x)$$
$$n_1 = \sum C_{fg}(x)$$
$$k_2 = C_{bg}(x)$$
$$n_2 = \sum C_{bg}(x)$$

Where the suffixes fg and bg correspond to foreground and background, respectively. And now, if phraseness score and informativeness score are represented by φ_p and φ_i then the total score is

$$\varphi = \frac{1}{1 + \exp(-a\varphi_p - b\varphi_i + c)} \qquad (3)$$

Where a, b and c are estimated under user supervision.

Here an exponential equation is used instead of a linear one to combine the phraseness and informativeness scores because the BLRT score has a tendency of getting centralized around the most frequent words and therefore may vary in a large range depending on the data to be processed[2].

The following Eq. was derived and used to predict the occurrence of keywords in a phrase.

$$P(x, y) = \frac{f(x,y)}{f(x,y) + f(\sim x,y) + f(x,\sim y)} \qquad (4)$$

Where

P (x, y) = the probability of getting y after x,
f (x, y) = frequency of y occurring after x
f (~x, y) = frequency of y occurred with no x in front
f (x, ~y) = frequency of x with no y after it

The construction of the graph depends on the probability found in the above equation to construct a set of most likely phrases to occur after a given phrase. Now if we have the next phrase as one from the set, it is surely a relevant keyphrase. And in other cases, it needs to be judged by the statistical methods using the phraseness and informativeness scores.

The training algorithm: The system was trained using the probabilistic prediction approach. Algorithm 1 outlines the approach used for training the system. The algorithm's principal focus is on building the association graph.

Algorithm 1: BuildAssociationGraph
Inputs: L, the list of found keyphrases
Outputs: G, the graph representing the association

1. Set G←Null
2. For Each Element S in L Do
   Set n←Number of words in S
   2.1. For i←1 to n-1 Do
      2.1.1. Set X←ith word in S
      2.1.2. Set Y←i+1th word in S
      2.1.3. Compute p←p(x,y) from Equation 3.1
      2.1.4. If X or Y or Both are not included in G Then
      2.1.4.1. Add the absent node to G
      2.1.5. End if
      2.1.6. Set An Edge from X to Y with cost p in G
   2.2. End for
   3. End for
   4. Return G

THE PROCESSING ALGORITHM

For processing, a new algorithm was developed that decides its tasks based on the probabilistic index. Algorithm 2 specifies the algorithm for processing.

Algorithm 2: The processing algorithm
Procedure Extract Keyphrase (Document d)
1. Initialize S = Ø, P = Ø
2. For each word w in Document
   2.1. If w is in S
      2.1.1. Add w to P
      2.1.2. Add possible next keywords to S
   2.2. Else
      2.2.1. Save P
      2.2.2. Set P = Ø
   2.2.3. Calculate the scores of w
   2.2.4. If w can be considered as a Keyphrase or part of it
      2.2.4.1. Add w to P
      2.2.4.2. Set S to Possible next keywords
   2.2.5. End If
   2.3. End If
3. End

For steps 2.1.2 and 2.2.4.2. of Algorithm 2, the possible next keyphrases are identified on a threshold from the association graph built on the training phase.

PERFORMANCE ANALYSIS

This approach reduces time needed to compute the keyphrase and in most cases, identifies the keyphrases correctly after being trained well by the training documents. This approach is useful when the corpus leads to a specific domain. In former approach, for each keyphrase, the informativeness and phraseness scores needed to be computed and then the phrase was identified as a keyphrase or not. The same approach is used here with an extension that if a word remains in the list of predicted words, it is identified as a keyphrase without calculating the scores.

Computational time: The computational time can be considered on two phases, the training time and the processing time. Among them, training time is not likely to put much effect on the outcome. But processing time would matter more.

For the analysis of complexities regarding the predictive approach, the following things are considered

Total number of different words in the Knowledge Base = W
Total number of predictable keyphrases derived from knowledge= n
Average number of keywords in a keyphrase = m
Number of predictable keyphrases = k
Number of unpredictable keyphrases = p
Remaining words in the document = r
Total number of recognized keyphrases in the current document = N

For the k keyphrases that can be predicted from previously constructed graph, C_P =k * Time needed to locate the first word * (m-1) * time needed to locate the next (m-1) words.

$$C_P = k*\{O(\log_2 n)*(m-1)O(\log_2(m-1))\}$$

$$C_P = k(m-1)O(\log_2 n)O(\log_2(m-1)) \quad (5)$$

Now for the p keyphrases that could not be predicted from the previous knowledge, C_U = p * Time needed to compute phraseness and informativeness scores of each pair of m words.

$$C_U = p * \{m * O(\log_2 W) + (r-1)O(W^2)\}$$

$$C_U = pmO(\log_2 W) + p(r-1)O(W^2) \qquad (6)$$

Therefore, the final complexity for all recognizable keyphrases for the system would be

$$C = \frac{k}{k+p}C_P + \frac{p}{k+p}C_U \qquad (7)$$

It is apparent that if k = p, $C_P < C_U$ because in all cases, n, m<<W.

Therefore, it can be concluded that, the system would perform better as the number of predictable keyphrases increase. If the number of predictable keyphrases were zero, then the system would perform in the same complexity as the previous ones, that is, C_U.

In our processing, it is found that in different documents of same domain, 40% to 60% keyphrases can be predicted by this approach.

As $C_P < C_U$, it is apparent that the complexity is reduced as more keyphrase are predictable from the training phase and from the already processed documents. The only case where no keyphrases may be predicted is the case when a document of a different domain is processed for a system trained in a domain.

Computational efficiency: The system worked efficiently for most cases and it was found from experiment that it identifies the keyphrases more accurately as either the size of training cases or the size of processing cases increase.

The efficiency of the system is measured using the three well known Information Extraction Performance Measures. Namely, the F-measure, Precision and Recall. The definition for the values are

$$p = \frac{c}{n}$$

$$r = \frac{c}{m}$$

$$F = \frac{2pr}{p+r}$$

here,

p = precision r = recall F = F-measure

c = number of present values correctly predicted

n = number of values predicted to be present

m = number of present values

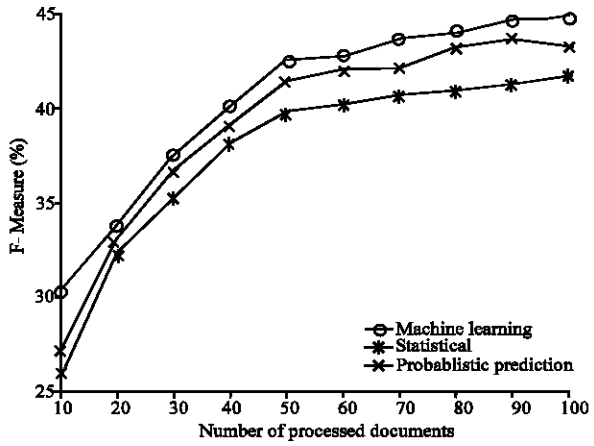
Along with these values, the term accuracy was used to describe the percentage of the keyphrases predicted in accordance with human extraction.

EXPERIMENTAL RESULTS

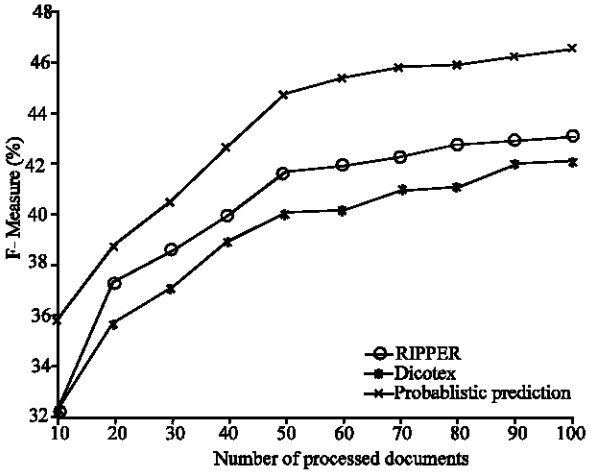
The approach is tested for a number of documents on different domains and yielded a result that gets better with more processing and also better with better number of training documents.

Generally, in fields where the keyphrase extraction system is used, 25-30 % data is repeated in processing which may mean either the inclusion of some documents from the training phase to the processing phase or repeating some processing documents or both.

The system was found to be performing with 58.2% average precision and 30.75% average recall yielding an



Data with non repeated components



Data with repeated components

Fig. 4: Comparison to other systems

average F-measure of 40.19% for the cases where part of data were not repeated. And if the part of data was repeated, it was found to be working with 62.71% average precision, 32.96% average recall yielding an average F-measure of 43.19%.

The comparison to the system with existing systems based on Machine Learning and Statistical Approach is shown on Fig. 2. The data on processing was collected using an Implementation of the statistical approach and another implementation of machine learning.

It is apparent that the system performs better when part of the data is repeated. In case of non-repeated data, the system performs between the performance levels of the existing approaches.

CONCLUSION

The predictive approach proposed in this paper has an intention to reduce the time needed to identify the keyphrases in a document and also to act as a tool to create new formations of keyphrases from the old ones. This approach is well suited for the documents representing a certain Domain. Technical papers on a field can be considered as ideal processing elements for this system. Future studies on this topic can work with using a more effective method than the graph to represent the association and ordering between the phrases.

REFERENCES

1. Turney, P.D., 2003. Coherent Keyphrase Extraction Via Web Mining, Proceedings of International Joint Conference on Artificial Intelligence.
2. Tomokiyo, T. and M. Hurst, 2003. A Language Model Approach to Keyphrase Extraction, ACL - 2003 workshop on Multiword Expressions : Analysis, Acquisition and Treatment.
3. Witten, I.H., G.W. Paynter, E. Frank, C. Gutwin, C.G. Neville Manning, 1999. KEA:Practical Automatic Keyphrase Extraction, proceedings of Digital libraries 99, ACM Press.