Context Based Indexing in Text Summarization Using Lexical Association

[1] Dipti D. Pawar
[1] M.Tech. Student, Bharati Vidyapeeth COEP,
Department of Computer Engineering,
Bharati Vidyapeeth College of Engineering,
Pune, India

[2]Prof. M. S. Bewoor and [3]Dr. S. H. Patil
[2]Asst. Professor, Department of Computer Engineering,
[3]Head, Computer Engineering Department,
Bharati Vidyapeeth College of Engineering,
Pune, India

Abstract

World Wide Web today is a largest source of online-information. Great amount of information is present on internet in the form of web pages. There has been a huge amount of work on query specific summarization of documents using similarity measure. Indexing weights of the document terms are utilized to compute the sentence similarity value which remains independent on context. Very little work has been done for the problem of context independent document indexing for the text summarization task. The main contribution of this research work is to combine both approaches of Lexical association and context sensitive indexing. While doing so we have also used novel concept of Lexical association between terms to measure the similarity between sentences using computed indexing Weights. The proposed concept of sentence similarity measure has been used with the graph-based ranking method to create document graph and obtain summary of document.

1. "Introduction"

Text Summarization is the method of condensing the input text into a shorter version, preserving its information content and overall meaning [1].This paper focuses on sentence extraction based text summarization. Most of the previous methods on the sentence extraction-based text summarization task use the graph based algorithm [2] to calculate importance of each sentence in document and most important sentences are extracted to generate document summary. These extraction based text summarization methods give an indexing weight to the document terms to compute the similarity values between sentences .Document features like term frequency, text length are used to assign indexing weight to terms. Therefore document indexing weight remains independent on context in which document term appears.

This proposed method aims at providing novel idea of context sensitive document indexing to resolve problem of context independent document indexing. Every document contains content-specific and background terms .The indexing methods used in existing models cannot differentiate between Terms reflected in sentence similarity values.

In this proposed method we are considering the problem of context independent document indexing using Lexical association. In the document, the content specific words will be highly related with each other, while the background terms will have very low relationship with the other terms in the document. In this research work the relation between document terms is captured by the lexical association, computed through a corpus analysis.

Context based document indexing is implemented using Page Rank-based algorithm [3] to compute how informative is each of the document term .Main motivation behind using Lexical association is the main assumption that the context in which word appears gives valuable information about its meaning [11]. Sentence similarity values calculated using the context sensitive indexing provides the contextual similarity between two sentences. This will allow two sentences to have distinct similarity values depending on the context.

2. "Background"

Text Summarization is information extraction technique, which generates shorter version or summary of original text such that generated summery is useful to give an overview of original text within a short duration. Text summary can be generic or query dependent. Text Summarization methods can be classified into extractive and abstractive summarization [1]. An extractive summarization method consists of selecting significant words, sentences, paragraphs etc. from the input document and concatenating them into shorter form. The significance of sentences is decided based on statistical and linguistic features of sentences. An Abstractive summarization attempts to develop an understanding of the important concepts in a document and then express those concepts in natural language.

Furthermore text Summarization can also be specific to the information needs of the user called as query specific document summarization [5].The task of producing summary from multiple documents is called multiple document summarizations [6].

Furthermore clustering based approach [7] can be use for text summarization that groups first, the similar documents into clusters & then sentences from every document cluster are clustered into sentence clusters. And top scoring sentences from sentence clusters are chosen in to the final summary. During the process of text summarization different condensation operations can be applied on entities such as words, phrases, clauses and sentences. These entities can be analyzed at various linguistic levels: morphological, syntactic, and semantic and discourse. Based on the level of linguistic analysis of the source, summarization methods [1] can be classified into approaches as follows

A) Shallow approach (Surface level analysis):

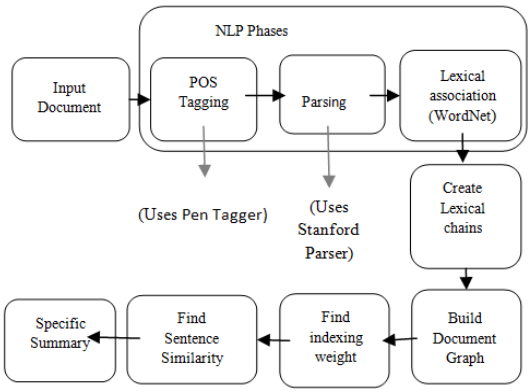
It considered features such as word count, presence of cue phrases, Position of the sentence to compute the important concepts of the document and saliency of the information.

B) Deeper approach (Deep semantic analysis):

It helps to find the theme or context of the content. Lexical association comes under this approach. It is the method to hold the text together by considering the semantic or identical relations between the words of the text.

3. "System Description"

There are several stages while generating summary as shown in Figure 1.



Description of each of stage is given in the following section:

3.1. Input

Initially original document containing set of statements is fed to the system. The stop words like "a", "her" and "has" which do not contribute to understanding the key ideas present in the text can be eliminated.

3.2. Splitting and Tokenization

Finding the end of a sentence is not an easy job for a computer. The input text is divided into separate sentences by the new line character and transferred into the array of paragraphs by using split method.

Tokenization is a method to separate the input text into separate tokens. The text can be separated into tokens. Punctuation marks, spaces and word terminators are the word breaking characters.

3.3. Part of Speech Tagging

Once the words are tokenized next POS tagger is applied to them for knowing their grammatical semantics. In POS tagging, we are classifying word under English Grammar Phrases.

CC- coordinating conjunction
VBN: Verb, past participle
CD- cardinal number
IS- existential there
IN: Preposition
DT- Determiner
NNP: Proper noun
NN- Noun
JJ- Adjective
DD-Common determinant

3.4. Text Chunking

Text chunking is dividing text into parts of words and forming groups like verb group, noun group. Each word is assigned only one unique tag. Text chunking can be defined as the method of breaking the sentence into a set of non-overlapping chunks. For e.g.

Sentence: I start with a perception: when I examine a segment, I examine it a chunk at a time.

Chunk: [I start] [with a perception]: [when I examine] [a segment], [I examine it] [a chunk] [at a time].

3.5. Parsing

In this research work we are using Stanford parser to create the parse tree for given sentence. Stanford parser is freely available on Internet It converts an input sentence into a hierarchical structure that corresponds to the units of meaning in the sentence. It can convert various forms of plain text into various analyzed formats.

3.6. Lexical Association

Our method takes advantage of the lexical association structure in the text in order to evaluate importance of sentences. It is used to capture theme or context of text. It is the method to hold the text together by

considering the semantic or identical relations between the terms of the text. The lexical association structure of the text can be modeled with the help of lexical chains. Lexical chains [8] are pattern of semantically related words, spanning over the entire text. A lexical chaining algorithm needs an ontology to obtain the semantic relations between word senses.WordNet (Fellbaum, 1998) is such an ontology, which is used by our method. In this proposed method we are computing lexical chains in two steps.

In first step metachains will be created .When a noun is encountered, for every sense of the noun in Word Net, the noun sense is placed into every metachain for which it has an identity, synonym, or hyperonym relation with that sense. These metachains shows all possible interpretation of the input text.

In the second step, we are finding the best interpretation, is determined by making a second pass through the input document. For each noun, each metachain to which the noun belongs is examined and a decision is made, based on the type of relation and distance measure, as to which metachain the noun contributes to most. The noun is then deleted from all other metachains.Once all of the nouns have been processed, we get the interpretation whose score is most. From this interpretation, the best scored chains can be selected.

In this research work we are using following parameters for deciding the strength of lexical chain.

i) Size: The number of occurrences of entities of the chain

ii) Distinct Entities: is the number of the distinct entities of the lexical chain.

$$Score\ (Chain) = Length * Homogeneity \quad , \qquad (1)$$

$$Homogeneity = 1 - \frac{\#Distinct\ Entities}{Size\ (chain)} \quad , \qquad (2)$$

Using Equation (1) and (2) from [7] we will calculate the score of each chain.

3.7. Find Context based Indexing Weight

The internet contains huge amount of electronic collections that always contain high quality information. The basic aim is to retrieve the top collection of information for a specific information need. Starting from a collection of unstructured documents, the indexing was used to retrieves a great amount of information like the list of documents, which contain a given document term. It also keeps record of number of all the occurrences of each document term within every document. This information is maintained in an index, which is called as an inverted file (IF).

The index consists of an array of the posting lists where each posting list is associated with a term as well as the identifiers of the documents containing the term. The term based Index seems to be less efficient due to information retrieval problems like Polysemy and Synonymy .Thus the importance of term for generating the index is reduced and the stress is given on the context of the document. Context provides extra information to help improve search result relevance.

Text summarization is the one of the application of information retrieval. It is very helpful for user to use context based indexing, when user needs a context specific summary from document containing more than one context or from multiple documents with different contexts. The context based index is consist of three fields, the one containing the context, the second one containing terms related to the context and the third one contains the lists of documents that contain the term with that specific context. After creation of lexical chains we can easily understand semantic relationships between words and the context of input document.

Next step is to calculate the context based indexing weight of each term using graph based Page ranking algorithm. In this graph for given document is built. Let G= (V, E) be an undirected weighted graph to reflect relationship between terms in document, where each vertex $V = \{vj \mid 1 \leq j \leq |V|\}$ denotes set of vertices and each vertex is document term and E is a matrix of dimensions $|V| \times |V|$. Each edge ejk € E gives the lexical association value between the terms corresponding to the vertices vj and vk. The lexical association between the same terms is set to 0. For implementation, the indexing weight of all terms is initially set to 1.0. Indexing weight of each term shows the importance of that term in document. The convergence of the recursive algorithm is achieved when the difference between the scores computed at two successive iterations falls below a given threshold ε.

3.8. Sentence similarity using indexing weights of terms

Next step is to find Similarity between sentences using the function sim (Si, Sj). Similarity values calculated using context based indexing weights of document terms reflects the contextual similarity between terms. In this for each sentence $\vec{Sj}$ in the document, the sentence vector Sj is built using calculated indexing weights of sentences. The sentence vector is calculated such that if a term vt present in sentence Sj, it is given a weight of term vt; else it is given a weight 0. The similarity between two sentences Si and Sj is computed using Equation (3).

$$Sim\ (Si, Sj) = \vec{Si}.\vec{Sj} \quad , \qquad (3)$$

At the end depending on the contextual similarity value summary will be generated specific to users query.

4. "Related work"

R.Varadarajan and V. Hristidis [5] developed a model to create Query Specific Summaries by identifying the most query-specific fragments and combining them using the semantic associations in the document. They focused on keyword queries since keyword search was the most popular information discovery method on documents. In particular, initially structure was added to the documents in the preprocessing stage and converted them to document graphs. Document graph was used to represent the hidden semantic structure of the document and then perform keyword proximity search on this graph. Then, the effective summaries were computed by calculating the top spanning trees on the document graphs.

R. Mihalcea [3] introduced a novel unsupervised Method for automatic sentence extraction using graph based ranking algorithms. The graph-based ranking algorithm is a method of deciding the importance of a vertex within a graph, by taking into account global information rather than only the local vertex-specific information. A similar method was applied to lexical or semantic graphs extracted from natural language documents using a graph-based ranking model called as Text Rank, which can be used for a variety of NLP applications where knowledge taken from a whole text was used in making local ranking decisions. Such text-based ranking methods can be applied to tasks ranging from automated extraction of key phrases, to extractive summarization and WSD. Text Rank finds the connections between various entities in a text, and executes the concept of recommendation.

R. Barzilay and M. Elhadad [8] introduced a new method to compute lexical chains in a text using knowledge sources like Word Net thesaurus, a part-of-speech tagger, and shallow parser. Summarization works in steps like Text segmentation, Lexical chain creation, scoring chains and sentence extraction. Furthermore they expand the set of candidate words to include noun compounds and evaluating importance of noun compounds by taking into account the noun compounds explicitly present in Word-Net. They generate chains in every segment using relatedness criteria, and next, they combine the chains from the different segments using much stronger criteria for connectedness only two chains are combined across a segment boundary only if they include a common word with the same meaning. For each text, they manually ranked chains in terms of relevance to the main topics. They then computed different parameters on the chains, including chain length, homogeneity index.

Erkan and Radev [10] introduced a stochastic graph-based method for computing relative importance of textual units for NLP. LexRank is use for calculating sentence significance based on the notion of eigenvector centrality in a graphical representation of sentences. In this model, a connectivity matrix based on intra-sentence cosine similarity was used as the adjacency matrix of the graph representation of sentences to assess the centrality of each sentence in a cluster and extract the most significant ones to include in the summary. They introduced different ways of defining the lexical centrality principle in multiple-document summarization, which computes centrality in terms of lexical characteristics of the sentences.

C. Zhou, W. Ding and Na Yang [9] authors introduced a dual indexing method for search engines based on campus Net. Indexing method, which means, it has document index as well as word index. Document index is depends on the documents to do the clustering, and arranged by the position in each document. In the information retrieval, the search engine first takes the document id of the word in the word index, and then goes to the position of respective word in the document index, because in the document index the word in the same text document is adjacent. The search engine compares the biggest word matching assembly with the sentence that user provides. The method proposed by them seems to be time consuming as the index exists at two levels.

5. "Conclusion and Future work"

In this work we presented an indexing structure that can be constructed on the basis of the context of the document. The context of the document can be extracted by using thesaurus and ontology repository. So this paper uses Lexical association for context based index building. The context based indexing enables extraction from index on the basis of context rather than keywords. This aids in improving the quality of the retrieved results. The context based indexing plays an important role in result space and time consumption. We show with a user survey that our approach performs better than other state of the art approaches.

In the future, we plan to extend our work to account for links between documents of the dataset. Also we will try to implement same algorithm in different applications. Furthermore same technique can be applied on different file formats and best indexing method can be suggested for different file formats.

References

[1]V.Gupta and G. S. Lehal ,A Survey ofText Summarization Extractive techniques, Journal of Emerging Technologies in Web Intelligence, Vol. 2, No. 3, August 2010.

[2]X. Wan and J. Xiao, Exploiting Neighborhood Knowledge for Single Document Summarization and Keyphrase Extraction, ACM Trans. Information Systems,vol.28, pp.8:1-8:34,http://doi. acm.org/10.1145/1740592.1740596, June 2010.

[3]R. Mihalcea , Graph-based Ranking Algorithms for Sentence Extraction, Applied to Text Summarization, Department of Computer Science University of North Tex asrada@cs.unt.edu.

[4]G. Ercan and I. Cicekli, Lexical Cohesion Based Topic Modeling for Summarization, Dept. of Computer Engineering Bilkent University, Ankara, Turkey.

[5] R. Varadarajan and V. Hristidis, A System for Query-Specific Document Summarization, School of Computing and Information Sciences Florida International University Miami, FL 33199.

[6]D. Radev, H. Jing , M. Sty's ,and D. Tam, Centroid-based summarization of multiple documents, Information Processing and Management 40 (2004) 919–938, University of Michigan, Ann Arbor, MI 48109, USA IBM T.J. Watson Research Centre, Yorktown Heights, NY 10598, USA, 24 October 2003.

[7] D. Sureshrao, S. Subhash and P. Dashore, Analysis of Query Dependent Summarization Using Clustering Techniques, International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume 2, Issue 1.

[8]R. Barzilay and M. Elhadad, Using Lexical Chains for Text Summarization, Mathematics and Computer

Science Dept. Ben Gurion University in the Negev Beer-Sheva, 84105 Israel .

[9]C. Zhou, W. Ding and Na Yang, Double Indexing Mechanism of Search Engine based on Campus Net, Proceedings of the 2006 IEEE Asia-Pacific Conference on Services Computing (APSCC'06) Quan, T. T., Hui, S. C., Fong, A.

[10]G. Erkan and D. Radev, LexRank: Graph-Based Lexical Centrality as Salience in Text Summarization, J. Artificial Intelligence Research,vol.22,pp.457-479http://portal.acm.org/citation.cfm?id=1622487.1622501, Dec. 2004.

[11] Z. Harris, Mathematical Structures of Language, Wiley, 1968.