WordNet-based Summarization of Unstructured Document

CHENGHUA DANG
Hebei University of Engineering
Handan, Hebei, 056038
PEOPLE'S REPUBLIC OF CHINA
dangchhua@yahoo.com.cn

XINJUN LUO
Hebei University of Engineering
Handan, Hebei, 056038
PEOPLE'S REPUBLIC OF CHINA
xjluoOZ@yahoo.com

HAIBIN ZHANG
Handan College
Handan, Hebei, 056030
PEOPLE'S REPUBLIC OF CHINA
chinesecobra2002@yahoo.com.cn

Abstract: - This paper presents an improved and practical approach to automatically summarizing unstructured document by extracting the most relevant sentences from plain text or html version of original document. This technique proposed is based upon Key Sentences using statistical method and WordNet. Experimental results show that our approach compares favourably to a commercial text summarizer, and some refinement techniques improves the summarization quality significantly.

Key-words: Document Summarization, Key Sentence, WordNet, POS, Semantic Similarity

1 Introduction

Text summarization is the process of condensing a source text while preserving its information content and maintaining readability. As the amount of information available in electronic format continues to grow, research into automatic document summarization has taken on renewed interest.

A summary can be employed in an indicative way – as a pointer to some parts of the original document, or in an informative way – to cover all relevant information of the text [1]. In both cases the most important advantage of using a summary is its reduced reading time. Technology of automatic summarization of text is maturing and may provide a solution to this problem [2, 3]. Automatic text summarization produces a concise summary by abstraction or extraction of important text using statistical approaches [4], linguistic approaches [5] or combination of the two [3, 6, 7]. In this paper, a practical approach is proposed for extracting the most relevant sentences from the original document to form a summary. The idea of our approach is to find out key sentences from the Keyword extraction based on statistics and Synsets extraction using WordNet. These two properties can be combined and tuned for ranking and extracting sentences to generate a list of candidates of key sentences. Then semantic similarity analysis is conducted between candidates of key sentences to reduce the redundancy. We provide experimental evidence that our approach achieves reasonable performance compared with a commercial text summarizer (Microsoft Word summarizer).

2 Related Work

2.1 Summarization Techniques

Text summarization by extraction can employ various levels of granularity, e.g., keyword, sentence, or paragraph.

MEAD [8], a state of the art sentence-extractor and a top performer at DUC, aims to extracts sentences central to the overall topic of a document. The system employs (1) a centroid score representing the centrality of a sentence to the overall document, (2) a position score which is inversely proportional to the position of a sentence in the

document, and (3) an overlap-with-first score which is the inner product of the tf * idf with the first sentence of the document. MEAD attempts to reduce summary redundancy by eliminating sentences above a similarity threshold parameter.
Other approaches for sentence extraction include NLP methods [9, 10] and machine-learning techniques [11, 12]. These approaches tend to be computationally expensive and genre-dependent even though they are typically based on the more general tf * idf framework. Work on generative algorithms includes sentence compression [13], sentence fusion [14], and sentence modification [15].

2.2 Keywords Extraction Techniques

Traditionally, keywords are extracted from the documents in order to generate a summary. In this work, single keywords are extracted via statistical measures. Based on such keywords, the most significant sentences, which best describe the document, are retrieved.
Keyword extraction from a body of text relies on an evaluation of the importance of each candidate keyword [16]. A candidate keyword is considered a true keyword if and only if it occurs frequently in the document, i.e., the total frequency of occurrence is high. Of course, stop words like “the”, “a” etc are excluded.

2.3 WordNet in Text Classification

WordNet [17] is an online lexical reference system in which English nouns, verbs, adjectives and adverbs are grouped organized into synonym sets or synsets, each representing one underlying lexical concept. A synset is a set of synonyms (word forms that relate to the same word meaning) and two words are said to be synonyms if their mutual substitution does not alter the truth value of a given sentence in which they occur, in a given context. Noun synsets are related to each other through hypernymy (generalization), hyponymy (speciali-zation), holonymy (whole of) and meronymy (part of) relations. Of these, (hypernymy, hyponymy) and (meronymy, holonymy) are complementary pairs.
The verb and adjective synsets are very sparsely connected with each other. No relation is available between noun and verb synsets. However, 4500 adjective synsets are related to noun synsets with pertainyms (pertaining to) and attra (attributed with) relations.
Scott and Matwin [18] propose to deal with text classification within a mixed model where WordNet and machine learning are the main ingredients. This proposal explores the hypothesis that the incorporation of structured linguistic knowledge can aid (and guide) statistical inference in order to classify corpora. Other proposals have the same hybrid spirit in related areas: Rodriguez, Buenaga, Gómez-Hidalgo, Agudo [19] and Vorhees [20] use the WordNet ontology for Information Retrieval; Resnik [21] proposes another methodology that index corpora to WordNet with the goal of increasing the reliability of Information Retrieval results.
Scott and Matwin [18], however, use a machine learning algorithm elaborated for
WordNet (more specifically, over the relations of synonymy and hyperonymy). This aims to alter the text representation from a non-ordered set of words (bag-of-words) to a hyperonymy density structure.

2.4 POS Tagging Techniques

Keyword extraction is conducted by counting the frequency of occurrence of a word and its syntactic variants in the document to be analysed. However, a single word or its variants may occur many times in a single document in different senses or part of speech, which could lead ambiguity.
In order to improve the quality of Keyword selection, part-of-speech tagging (POS Tagging) is considered.
POS Tagging, also called grammatical tagging, is the process of marking up the words in a text as corresponding to a particular part of speech, based on both its definition, as well as its context - ie. relationship with adjacent and related words in a phrase, sentence, or paragraph. A simplified form of this is commonly taught school-age children, in the identification of words as nouns, verbs, adjectives, adverbs, etc.
The task of POS Tagging is to identify the correct part of speech (POS - like noun, verb, pronoun, adverb ...) of each word in the sentence. The algorithm takes a sentence as input and a specified tag set (a finite list of POS tags). The output is a single best POS tag for each word. There are two types of taggers: the first one attaches syntactic roles to each word (subject, object, ..) and the second one attaches only functional roles (noun, verb, ...). There is a lot of work that has been done on POS tagging. The tagger can be classified as rule-based or stochastic. Rule-based taggers use hand written rules to disambiguate tag ambiguity. An example of rule-based tagging is Brill's tagger (Eric Brill algorithm) [22]. Stochastic taggers resolve tagging ambiguities by using a training corpus to compute the probability of a given word having a given tag in a given context. For example: tagger using the Hidden Markov Model, Maximize likelihood.
There are quite a few of open-source POS Tagger available now, like GATE, which can be directly utilized and help reduce development work significantly.

2.5 GATE POS Tagger

GATE is a software architecture for language engineering, developed by The University of Sheffield. As an architecture, GATE suggests that the elements of software systems that process natural language can usefully be broken down into various types of component, known as resources [23]. Components are reusable software chunks with well-defined interfaces, and are a popular architectural form, used in Sun's Java Beans and Microsoft's .Net,

GATE utilizes the Hepple tagger [24], a modified version of the Brill tagger, to produce a part-of-speech tag as an annotation on each word or symbol.

The tagger uses a default lexicon and ruleset (the result of training on a large corpus taken from the Wall Street Journal). Both of these can be modified manually if necessary. Two additional lexicons exist - one for texts in all uppercase (lexicon cap), and one for texts in all lowercase (lexicon lower). To use these, the default lexicon should be replaced with the appropriate lexicon at load time. The default ruleset should still be used in this case.

2.6 Document Format Conversion

Techniques described above are only applied to documents in format of plain text. However not all documents are plain text, in fact, most are binary, like in Microsoft WORD or PDF or else. Document summarization only cares about the content, therefore converting non-plain-text document into plain text is necessary. There are a great number of commercial conversion tools available on market. We have been employing Cambridge xDoc [25] and Oracle Text [26] in our research.

Cambridge xDoc provides out-of-the-box support to convert Microsoft WORD or PDF files into HTML using pure Java and XML transformation technologies. With xDoc, WORD / PDF documents located on local machines or published on internet can be transformed in a multi-step manner. Firstly, xDoc reads a binary doc/pdf file, and convert it into a stylistic XML output that captures all of the document's content, styles, formatting, layout and graphics information. Secondly, xDoc transforms that stylistic XML into HTML, using provided, out-of-the-box stylesheets. In addition, because of the open XML-based approach that xDoc uses in transforming the Word/PDF documents, programmatic access to content is allowable before it gets transformed.

Oracle Text is another powerful tool for document format conversion. The automatic document filtering technology in Oracle Text enables to index up to more than 150 document formats. This technology also enables you to convert documents to HTML or plain text for document presentation with the CTX_DOC PL/SQL package. To use automatic filtering technology for converting documents to HTML with the CTX_DOC package, you need not use the AUTO_FILTER indexing preference, but you must still set up your environment to use this filtering technology.

Typically, a query application allows the user to view the documents returned by a query. The user selects a document from the hitlist and then the application presents the document in some form. With Oracle Text, you can display a document in different ways. For example, you can present documents with query terms highlighted. Highlighted query terms can be either the words of a word query or the themes of an ABOUT query in English. You can generate three types of output associated with highlighting: a marked-up version of the document, a plain text version of the document (filtered output), and highlight offset information for the document. The three types of output are generated by three different procedures in the CTX_DOC (document services) PL/SQL package. In addition, you can obtain plain text and HTML versions for each type of output. You can also obtain gist (document summary) and theme information from documents with the CTX_DOC PL/SQL package.

With extensive comparison between the html outputs of xDoc and Oracle Text, Oracle Text was adopted by us to provide document format conversion service due to much better output in terms of presentation. In addition, Oracle Text also provides plain text version, and more importantly it is able to cater for up to 150 formats.

3 Our Algorithms

3.1 Preprocessing of the text

1) Convert unstructured document into plain text.
2) Break the text into sentences.
3) Stop-word elimination – common words with no semantics and which do not aggregate relevant information to the task (e.g., "the", "a") are eliminated;
4) POS tagging: produces a part-of-speech tag as an annotation on each word or symbol.

5) Case folding: consists of converting all the characters to the same kind of letter case - either upper case or lower case;
6) Stemming: syntactically similar words, such as plurals, verbal variations, etc. are considered similar; the purpose of this procedure is to obtain the stem or radix of each word, which emphasize its semantics.

3.2 Keyword Refinement using POS Tagger

In the extraction of Key Phrases [27, 28], noun or adjective phrases are considered. The use of noun and adjective is applied to our application as well, so the selection of Keywords is limited in noun or adjective words.
With POS Tagger, each word has been marked as noun, adjective or verb, etc. In the calculation of word occurrence, all non-noun and non-adjective variants are excluded. For example, "promotion" would be considered, while "promote" not.

3.3 Synsets Ranking

The basic motivation of this step is to rank the synsets based on their relevance to the text. So, if lots of words in the text correspond to the same synset, that synset or 'meaning' is more relevant to the text, and thus, it must get a higher rank. This idea has been borrowed from [28], which details the use of WordNet Synsets as a mode of text representation.

3.4 Keyword Refinement using Synsets

Collection of Keywords are refined as compared with Synsets obtained above. The comparison is conducted by calculating the similarity between Keywords and Synsets. According to the vectorial model, this feature is obtained by using the Synsets of the document as a "query" against all the Keywords of the document; then the similarity of the document's Synsets and each Keyword is computed by the cosine similarity measure [29]. Then we retain those Keywords which have the closest similarity to the Synsets.

3.5 Key Sentences Selection

3.5.1 Sentence Ranking

Once the keywords are identified, the most significant sentences for summary generation can be retrieved from all narrative paragraphs based on the presence of keywords [30]. The significance of a sentence is measured by calculating a weight value, which is the maximum of the weights for word clusters within the sentence. A word cluster is defined as a list of words which starts and ends with a keyword and less than 2 non-keywords must separate any two neighboring keywords [16]. The weight of a word cluster is computed by adding the weights of all keywords within the word cluster, and dividing this sum by the total number of keywords within the word cluster.
The weights of all sentences in all narrative text paragraphs are computed and the top Ten sentences (ranked according to sentence weight) are the candidates of key sentences to be included in the summary.

3.5.2 Refinement of Sentence Ranking

The top ten candidates of key sentences are selected from the text based upon their relevance to the document, however, we need to keep in mind that two sentences are similar to each other in terms of semantic content cannot be both selected. Semantic similarity [31, 32] between sentences is calculated to exclude those redundant candidates of key sentences.
Given two sentences, how similar the meaning of two sentences is can be determined by the measurement. The higher the score. the closer the meaning of two sentences is. Steps to measure semantic similarity between two sentences are:

- Tokenization of each candidate key sentence.
- POS tagging.
- Words stemming.
- Word sense disambiguation.
- At last calculation of similarity of sentences based on the similarity of the pairs of words.

Finally the overall summary is formed by the top 25 keywords and the top 5 key sentences which have least similarity in semantic content. These numbers are determined based on the fact that key sentences are more informative than keywords, and the whole summary should fit in a single page.

4 Experiments

Summaries can be evaluated using intrinsic or extrinsic measures [33]. While intrinsic methods attempt to measure summary quality using human evaluation thereof, extrinsic methods measure the same through a task-based performance measure such the information retrieval-oriented task.
Intrinsic approach was utilized in our experiments. However, it is a time-consuming process to identify important units in documents by humans, therefore, we chose the Microsoft Word summarizer of MS Office 2000 to output summary baselines.
The comparison between our algorithm and the summarization algorithm for MS Word 2000

demonstrates that our experimental results give the best summarization at around 35% summary of a document.
Moreover, the use of noun / adjective enhances the reliability of selected Keywords compared to those without. The overall quality of summarization based on use of noun / adjective is also improved by 25%.
Eventually, removal of redundant sentences using semantic similarity between candidates of key sentences makes an achievement of another 5% improvement in the overall quality of summarization.

5 Conclusion

In this paper a combined technique for the extraction of key-sentences from an unstructured document is presented. It requires no training and makes use of publicly available lexical resources only. Such sentences are taken as a summary of the same document. Refining key sentences against WordNet semantic similarity comprehensively improve the correctness of automatic summary since redundancy is reduced to the minimum.

Since CTX_DOC.GIST pl/sql package of Oracle Text is able to generate gists and theme summaries of documents to be indexed, it is worth making a comparison between results of our method and Oracle Text in the future.

References:

[1] Mani, I. Automatic Summarization. J.Benjamins Publ. Co. Amsterdam Philadelphia (2001).

[2] I. Mani and M. Maybury. Advances in Automatic Text Summarization. MIT Press, ISBN 0-`262-13359-8, 1999.

[3] I. Mani. Recent developments in text summarization. In ACM Conference on Information and Knowledge Management, CIKM'01, pages 529–531, 2001.

[4] O. Buyukkokten, H. Garcia-Molina, and A. Paepcke. Seeing the whole in parts: Text summarization for web browsing on handheld devices. In Proceedings of 10th International World-Wide Web Conference, 2001.

[5] C. Aone, M.E. Okurowski, J. Gorlinsky, and B. Larsen. A scalable summarization system using robust NLP. In Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summari-zation, pages 66–73, 1997.

[6] R. Barzilay and M. Elhadad. Using lexical chains for text summarization. In Proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS'97), ACL, Madrid, Spain, 1997.

[7] J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell. Summarizing text documents: Sentence selection and evaluation metrics. In Proceedings of SIGIR, pages 121–128, 1999.

[8] D. Radev, H. Jing, and M. Budzikowska. Centroid-based summa-rization of multiple documents: sentence extraction, utility-based evaluation and user studies. In ANLP/NAACL Workshop on Automatic Summarization, pages 21–29, 2000.

[9] C. Aone, M. E. Okurowski, J. Gorlinsky, and B. Larsen. A Scalable Summarization System Using Robust NLP. In Proceedings of the Intelligent Scalable Text Summarization Work-shop, pages 66–73, 1997.

[10] R. Barzilay and M. Elhadad. Using lexical chains for text summarization. In *Proceedings of the Intelligent Scalable Text Summarization Workshop*, pages 10–17, 1997.

[11] C. Nobata and S. Sekine. Results of CRL/NYU System at DUC-2003 and an Experiment on Division of Document Sets. In 2003 Document Understanding Conference Draft Papers, pages 79–85, 2003.

[12] S. Teufel and M. Moens. Sentence extraction as a classification task. In ACL/EACL Workshop on Intelligent and Scalable Text Summarization, 1997.

[13] C.-Y. Lin. Improving Summarization Performance by Sentence Compression - A Pilot Study. In Proceedings of the International Workshop on Informa-tion Retrieval with Asian Language, pages 1–8, 2003.

[14] K. Han, Y. Song, and H. Rim. KU Text Summarization System for DUC 2003. In Document Understanding Conference Draft Papers, pages 118–121, 2003.

[15] A. Nenkova, B. Schiffman, A. Schlaiker, S. Blair-Goldensohn, R. Barzilay, S. Sigelman, V. Hatzivassi-loglou, and K. McKeown. Columbia at the Document Understanding Confe-rence 2003. In 2003 Document Understanding Conference Draft Papers, pages 71–78, 2003.

[16] O. Buyukkokten, H. Garcia-Molina, and A. Paepcke. Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices. In Proceedings of Tenth International World Wide Web Conference, 652–662, 2001.

[17] Fellbaum, Christiane (Ed.), Wordnet : An Electronic Lexical Database (Language, Speech and Communica-tion). MIT Press, 1998.

[18] S. Scott and S. Matwin, Text classification using WordNet hyper-nyms. In "Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems", Montreal, 1998.

[19] Buenaga M. Rodríguez., J. M. Gómez-Hidalgo, B. Díaz Agudo, Using WordNet to complement training information in text categorization. In "Proceedings of the International Conference on Recent Advances in Natural Language Processing", Tzigov Chark, 1997.

[20] M. Vorhees, Ellen, Using WordNet for text retrieval. In Fellbaum C. (ed.) "WordNet: An Electronic Lexical Database", MIT Press, 1998.

[21] Resnik Philip, Using information content to evaluate semantic similarity in a taxonomy. In "Proceedings of the 14th International Joint Conference on Artificial Intelligence", Montreal, 1995.

[22] Brill, Eric. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. *Computa-tional Linguistics* 21(4), 543-566, December, 1995.

[23] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, C. Ursu, M. Dimitrov, M. Dowman, N. Aswani and I. Roberts. Developing Language Processing Components with GATE, Version 4 (a User Guide), Oct 2006.

[24] M. Hepple. Independence and commitment: Assumptions for rapid training and execution of rule-based POS taggers. In Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000), Hong Kong, Oct 2000.

[25] CambridgeDocs. http://www.cambridgedocs.com

[26] Oracle Text 9.0.1 Technical Overview. http://www.oracle.com/technology/products/text/x/Tech_Overviews/text_901.html

[27] Witten, Ian H., Gordon W. Paynter, Eibe Frank, Carl Gutwin & Craig G. Nevill-Manning. KEA: Practical Automatic Keyphrase Extraction. Proceedings of the FOURTH ACM Conference on Digital Libraries, 1999.

[28] Ken Barker and Nadia Cornacchia. Using Noun Phrase Heads to Extract Document Key Phrases**.** In the Proceedings of the 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence, 2000.

[29] Ramakrishnan and Bhattacharya, Text representation with wordnet synsets. Eight International Conference on Applications of Natural Language to Information Systems (NLDB2003), 2003.

[30] Salton, G.; Buckley, C. Term-weighting approaches in automatic text retrieval. Information Processing and Management 24, 513-523. 1988.

[31] Chuang, W., and Yang, J. Extracting Sentence Segments for Text Summarization: A Machine Learning Approach. In Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 152–159, 2000.

[32] Patwardhan, S., and Pederson T. Using WordNet-based Context Vectors to Estimate the Semantic Relatedness of Concepts. *Proceedings of the EACL 2006 Workshop Making Sense of Sense - Bringing Computational Linguistics and Psycholinguistics Together*. Trento, Italy, 1-8, April 2006.

[33] Wan S., and Angryk R. A. Measuring Semantic Similarity using WordNet-based Context Vectors. IEEE International Conference on Systems, Man and Cybernetics, ISIC, 2007.

[34] D. R. Radev, E. Hovy, and K. McKeown. Introduction to the special issue on summarization. Computa-tional Linguistics, 28(4):399–408, 2002.