

A SEARCH STRATEGY BASED ON KEYWORD GENERATION USING COREFERENCE RESOLUTION

Ram Gupta, Vineet Gupta
B.E., Final Year
Computer Sc. & Engg., MITS
Gwalior 474005
{ramgupta.cse, vineet.cs}@gmail.com

Abstract

World Wide Web makes up a global information system that is extremely large-scale, diverse and dynamic, albeit searching the relevant information is a challenging and time consuming task for a user. Thus the strength of WWW is its weakness. Internet search engines have been used to deal with such problems. The search strategy employed by current search engines ask user for a string input which forms the query for the search engine. This restricts users to put up their requirements in keywords. But most of the times users can't find keywords that best describe their search requirements. Thus finding relevant information becomes challenging and time consuming.

In this paper, we have presented a new search strategy that employs automatic keyword generation from the supplied document. Proposed strategy takes a document as input rather than a simple string of keywords. It then generates keywords automatically using coreference resolution technique that best describe document's content and searches for them. Thus enabling the user to concentrate on the information to be searched instead concentrating on keywords that best describe the information.

1 Introduction

World Wide Web, as the name indicates is a large web of interconnected information resources. The explosive growth of the WWW in recent years has dramatically increased the volume of text documents on the Internet. Every day hundreds of new text documents appear on Internet increasing already enormous amount of ac-

cessible text information. To address this problem, several tools have been developed to help search relevant information more effectively by either assisted browsing or keyword/phrase based searching. Assisted browsing, such as WebWathcer (Joachims et al., 1997) and Syskill & Webert (Pazzani et al., 1996), guides/suggests a user along an appropriate path through the web based on its knowledge of the user's interests, of the location and relevance of various items in the collection, and the way in which others have interacted with the collection in the past. Internet search engines, such as AltaVista and Lycos, sends out spiders or robots to index any visited web pages and allow keyword or phrase based search. One characteristic of these approaches is they all rely on keywords provided by the user related to the information domain they want to search. Thus keywords determine the relevancy of search results. But most of the times users can't find keywords that best describe their search requirements due to a variety of reasons like user is novel to Internet, they are not familiar with the jargon or there is no exact phrase that suitably identify their requirement. This makes these approaches either time consuming or the search results often not quite relevant to what a user wants.

Our searching strategy that employs automatic keyword generation using coreference resolution technique is based on the fact that a document can be categorized by a set of keywords that form the skeleton of the document's content. These keywords represent the hidden theme of the document not visible on its surface and thus remain unidentified by users. Enabling users by providing the suitable keywords freed them from the problem of identifying phrases for the information that they want to search. Instead burning their calories in keywords identification users can now concentrate on information to be searched.

Our proposed strategy cooperates with a Natural Language Processing (NLP) framework, named General Architecture for Text Engineering (GATE) (Cunningham et al., 2004) developed at University of Sheffield to parse the document and generate nominal and pronominal coreference. Coreference chains are then built up and longest ones are selected (Bergler et al., 2003) to extract the keywords out of the document and formatted as appropriate query to search engines such as Google or other search engines. While the idea of using the length of coreference chains is not novel to the summarization and keyword generation community (Brunn et al., 2001; Lal and Rueger, 2002), our approach is distinguished by its purity coupled with its implementation for searching: no other technique is used to identify keywords for the search query.

2 Keyword Generation Model

Extracting keywords from the document supplied by the user is the cumulative result of various phases that parse and annotate the document with POS tags and coreference resolution information. This requires a NLP framework that in our case is GATE. GATE is provided with a processing resource that extracts information from the natural language sentences, named ANNIE. It can take documents in variety of formats like plain text, html, xml and others and returns an annotated xml file that contains POS tags and coreference resolution information. Our module accepts this xml file to build coreference chains and finding the longest ones. The entities those are referred by these chains form the keywords for the search query and probably represent the most important contents of the document. The number of keywords generated can be controlled by having a curb on the longest coreference chain. The entire process can be summed up into following phases:

3 ANNIE Pre-Processing Pipeline

ANNIE stands for *A Nearly-New Information Extraction system*. It relies on finite state algorithms and the JAPE language¹. The lexicon and ruleset incorporated are the result of training on a

¹ JAPE allows recognition of regular expressions in annotations on documents. Its grammar consists of a set of phases, each of which consists of a set of pattern action rules.

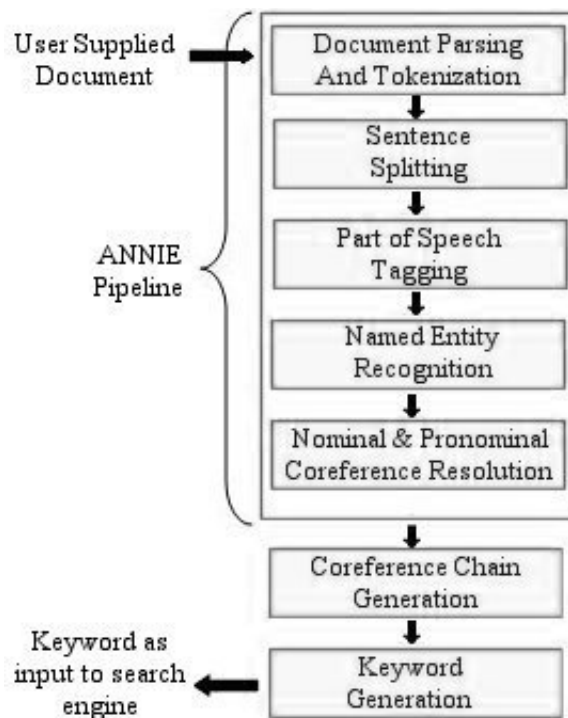


Figure 1: Shows various phases in keyword generation

large corpus taken from the Wall Street Journal. ANNIE pre-process the document supplied by the user, right from the tokenization to coreference resolution. ANNIE preprocessing pipeline comprises following processing resources:

- Document Reset PR
- English Tokeniser
- Gazetteer
- Sentence Splitter
- NE Transducer
- POS Tagger
- OrthoMatcher
- Nominal Coreferencer
- Pronominal Coreferencer

The output of this pipeline is an xml file which contains annotation that marks POS tags and coreference information. This tagged file forms the basis for the keyword generation module, which exploits the coreference information present in it.

4 Coreference Resolution

Coreference resolution is the process of determining whether two expressions in natural language refer to the same entity in the world. It is an important subtask in natural language processing systems. In particular, information extraction (IE) systems like those built in the DARPA Mes-

sage Understanding Conferences (Chinchor, 1998; Sundheim, 1995) have revealed that coreference resolution is such a critical component of IE systems that a separate coreference subtask has been defined and evaluated since MUC-6 (MUC-6 1995). Specifically, a coreference relation denotes an identity of reference and holds between two textual elements known as markables, which can be definite noun phrases, demonstrative noun phrases, proper names, appositives, sub-noun phrases that act as modifiers, pronouns, and so on.

4.1 Nominal Coreference Resolution

It is coreference resolution between proper nouns that occur in texts. This is accomplished with the help of gazetteer lists that identify proper nouns and annotate them with corresponding tags.

4.2 Pronominal Coreference Resolution

It is coreference resolution between pronouns that occur in texts. The pronominal coreference module performs anaphora resolution using the JAPE grammar formalism. The main module consists of three sub modules:

- quoted text module
- pleonastic it module
- pronominal resolution module

The module works according to the following algorithm:

- Preprocess the current document. This step locates the annotations that the submodule need (such as Sentence, Token, Person, etc.) and prepares the appropriate data structures for them.
- For each pronoun do the following:
 - inspect the proper appropriate context for all candidate antecedents for this kind of pronoun;
 - choose the best antecedent (if any);
- Create the coreference chains from the individual anaphor/antecedent pairs and the coreference information supplied by the OrthoMatcher (this step is performed from the main coreference module).

5 Keyword Generation

After the execution of ANNIE pipeline the resulting output is an xml file, containing annotations that mark information regarding POS tag-

ging and coreference resolution. Our module traverses this xml file to locate these annotations to build coreference chains. These chains are thence arranged in descending order to locate the longest ones. The entities those are referred by these longest chains form the keywords for the search query as the entities that occur more frequently and often represent the important information in the document. The relevancy of the search results is related to the number of the keywords generated, which is under the control of user. Our module works according to the following algorithm:

Algorithm KEY_GEN

Input: Annotated data (xml file)

K: Number of keywords

1. For each node do the following:
 - 1.1 If node has *antecedent_offset* then:
 - Get its value
 - 1.2 If no chain exists for that value then:
 - Create a new chain
 - 1.3 Add the value of node to the chain
2. Calculate the length of each chain generated
3. Arrange them in descending order of their length
4. Get the **K** longest chains

5. Return entities correspond to longest chain

Algorithm 1: Algorithm for keyword generation

5.1 A Walk-Through Example

Let's consider a paragraph to show how our model works:

Tom lives in Los Angeles. He is an actor. He likes it there. His wife Katie lives there, too. She doesn't like it so much. Katie and he have three cars. They are made by Toyota, Honda, and General Motors.

After running ANNIE pipeline following annotations are obtained:

Tom¹ lives in Los Angeles². He¹ is an actor³. He¹ likes it there. His¹ wife Katie⁴ lives there, too. She⁴ doesn't like it so much. Katie⁴ and he¹ have three cars. They are made by Toyota⁵, Honda⁶, and General Motors⁷.

- Tokens tagged with 1 refer to Tom.
- Tokens tagged with 2 refer to Los Angeles.
- Tokens tagged with 3 refer to *JobTitle* actor.
- Tokens tagged with 4 refer to Katie.
- Tokens tagged with 5, 6, 7 refer to Toyota, Honda and General Motors respectively.

After this we calculated the length of coreference chains. The following coreference chains were generated:

Coreference Chain	Length
Tom -> He -> He -> His -> he	5
Los Angeles	1
actor	1
Katie -> She -> Katie	3
Toyota	1
Honda	1
General Motors	1

Table 1: Coreference Chains

Finally Keyword Generation Algorithm is applied to the generated chains.

Constraint: Maximum no. of keywords = 2

Firstly the coreference chains are arranged in the descending order of their length. Then the tokens that correspond to the first two coreference chains are selected as keywords.

Keywords generated: Tom, Katie

Keywords generated are fed to the search engine that finally returns the search results.

6 Evaluation

We evaluated our model on an arbitrary web webpage. The page that we choose was about Mahatma Gandhi. The location of the page is: <http://www.sscnet.ucla.edu/southasia/History/Gandhi/gandhi.html>

The comparison was made with the search results provided by the advanced search facility of Google (Search Similar Pages). Following observations were made:

S. No.	Description of the Result
1.	History and Politics, Mahatma

	Gandhi
2.	Welcome to Mahatma Gandhi one spot complete information website
3.	Mahatma Gandhi Album: The Great Experimenter
4.	http://www.nuvs.com/ashram/
5.	Vajpayee, Atal Bihari on Encyclopedia.com
6.	South Asian Media Net
7.	South Asian Media Net
8.	webindia123-Indian Film personalities-mira nair
9.	The Official Mahatma Gandhi eArchive & Reference Library
10.	Mahatma Gandhi Indian Spiritual/Political Leader and Humanitarian

Table 2: Google Results

S. No.	Description of the Result
1.	The Official Mahatma Gandhi eArchive & Reference Library
2.	BBC - History - Mohandas Gandhi (1869 - 1948)
3.	History and Politics, Mahatma Gandhi
4.	SIR STAFFORD CRIPPS STATEMENT ON INDIA London, August 5, 1942 "The ...
5.	Mahatma Gandhi
6.	MAHATMA GANDHI - SOUTH AFRICA'S GIFT TO INDIA?
7.	Meet India's Sonia Gandhi by The Globalist - The Globalist ...
8.	Mahatma Gandhi Album: Gandhi: A Biography
9.	Mohandas Karamchand Gandhi, India (1869-1948) - Hall of Freedom ...
10.	Latest News by NewKerala, India

Table 3: Our Results

Analysis of the above results shows that:

- Only the initial 3-4 results of Google were relevant to the content of the test webpage.
- The rest were mostly irrelevant to the search content.
- In our case except the last one, all the results were mostly close to content of the test webpage.

7 Conclusion And Future Work

In this paper we have proposed a search strategy based on keyword generation using coreference resolution. We started with introduction of AN-NIE which is a processing resource incorporated in GATE. The approach presented here achieves comparable performance without using any heuristic approach. This shows that even simple coreference rules can be sufficient for tasks like keyword generation which can be used for searching similar type of documents. Particularly, we proposed a keyword generation algorithm that can extract relevant and frequent entities from a given document that form the keywords for the search query. This enables a user to search web pages similar to the document considered. The experimental results have proved the effectiveness of our approach.

Furthermore, we'd like to incorporate the text summarization onto the search results produced; so that the user can have an overview of the contents of the searched document, before actually visiting that web-site. We'd also like to use WordNet so that our model can also search the terms/synonyms related to the keywords generated. These features may be helpful to improve the search relevancy.

References

- Beth M. Sundheim, 1995. "Overview of results of the MUC-6 evaluation". In Proceedings of the Sixth Message Understanding Conference (MUC-6), pages 13-31.
- Hamish Cunningham et al., 2004. "GATE, a general architecture for text engineering". The University of Sheffield, <http://gate.ac.uk>.
- Meru Brunn, Yllias Chali, and Christopher J. Pinchak, 2001. "Text summarization using lexical chains". In Document Understanding Conference (DUC), New Orleans, Louisiana, USA, September 13-14, 2001.
- Michael Pazzani, Jack Muramatsu & Daniel Billsus, "Syskill & Webert :Identifying interesting Web Sites", 1996. In Proceedings of the Thirteenth National Conference on Artificial Intelligence, pp. 5461, Portland, 1996.
- Nancy A. Chinchor, 1998. "Overview of MUC-7/MET-2". In Proceedings of the Seventh Message Understanding Conference(MUC-7). <http://www.itl.nist.gov/iad/894.02/related-projects/muc/proceedings/muc-7-toc.html>.
- Partha Lal, Stefan Rueger. 2002. "Extract-based summarization with simplification". In NIST, 2002 (NIS, 2002).
- Sabine Bergler et al., 2003. "Using Knowledge-poor Coreference Resolution for Text Summarization", Concordia University, Canada.
- Thorsten Joachims, Dayne Freitag and Tom M. Mitchell, 1997. "Web-Watcher: A Tour Guide for the World Wide Web," In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, 1997.