Automatic Summarization of Arabic Texts

Jawad Berri and Omar Alghafri
Emirates Telecommunications Corporation
Etisalat University College
Department of Computer Engineering
P.O. Box 573 Sharjah
United Arab Emirates
Tel: (971 6) 5043585
Fax: (971 6) 5611789
berri@ece.ac.ae
http://www.ece.ac.ae

Abstract

This paper presents an automatic summarizer for Arabic texts. The main goal of a summarizer is to produce a condensed version of the content of an input text for various users. The approach exploits mainly the expressions of the language present in the text denoting sentences' relevance according to the author's point of view regardless of the domain. The summarizer selects the most relevant sentences based on linguistic knowledge in the form of linguistic patterns representing language expressions and word lists. Language expressions express the language knowledge and are independent from any specific domain. The paper presents the linguistic acquisition process which feeds the linguistic knowledge base, the architecture of the system including three modules, and the system implementation.

Key-Words

Automatic summarization, knowledge-based system, linguistic knowledge, linguistic pattern, semantic labels, matching algorithm.

1. Introduction

On-line document sources, such as the Internet, provide nowadays huge amounts of daily information destined to various users. In order to help users find useful information, search engines have been made available. However, as information is increasing, search engines retrieve a quantity of documents that need to be analyzed in order to select the needed information. Automatic summarizers seem to be good candidates for this task as they produce a condensed version of the original text to the comfort of the user who will be able to decide on the relevance of the original document.

Along with various approaches that have been used, we have developed a linguistic knowledge based approach [1, 2, 3] that uses mainly linguistic knowledge to summarize texts. Our approach exploits the expressions of the language present in the text denoting the relevance according to the author's point of view regardless of the domain.

The summarizer presented in this paper uses a linguistic knowledge base including linguistic patterns mapping language expressions and word lists to summarize Arabic texts. The architecture of the system includes three modules and a set of language processing tools that allow processing, identifying and then a selection of relevant sentences of a text based on their content.

The remaining sections of this paper are organized as follows: Section two gives an overview of approaches used in automatic summarization. Section three presents the knowledge acquisition of linguistic knowledge. In section four we present the architecture and the implementation of our system. Finally, section five concludes this paper with future directions to extend this work.

2. Summarization Approaches (Related Work)

Since the paper of Luhn [4] dating more than forty years back various approaches have been proposed and used to develop summarization systems [3, 5]. These approaches have emanated from different disciplines hence using different methods. For disciplines such as artificial intelligence and cognitive psychology "*summarization is understanding*", the system needs to understand the text to be able to summarize it [6, 7, 8]. These approaches require a representation of the text content, which is then reduced to the minimum, and finally a summary is generated from the reduced representation. These approaches are supposed to deliver high quality summaries comparable to those produced by the human however; it necessitates huge language resources and sophisticated natural language processing tools capable of analyzing and representing subtle details of the language. Most of the systems developed within these approaches are laboratory restricted; they are able to process generally texts related to a specific domain.

As an alternative to the above approaches, natural language engineering which aims to solve a variety of real-life problems, offers more practical approaches. They consist in extracting the most relevant sentences of a document in order to provide a summary. The objective is to implement systems capable of producing summaries suitable for targeted tasks and users. These approaches are more appropriate to process large amounts of texts in different domains. In this section we describe three summarization approaches namely *statistical-based*, *discourse structure-based*, and *linguistic knowledge-based* approaches.

Statistical-based approach relies mainly on the selection of sentences based on term frequency [4, 9]. The method uses some linguistic knowledge to identify a list of *stop words* ("words that carry no significance", e.g. conjunctions, prepositions, connectors) that are to be excluded from the frequency calculation, and possibly identifies in addition a list of *cue phrases* (for instance "significant", "impossible", "hardly") to be used to support the selection of sentences. Not unlike information retrieval systems [10, 11], statistical-based approaches are suitable to deal with relatively large amounts of texts however they deny absolutely any semantic in the analysis of the user's query as well as the text content.

Discourse Structure-based approach aims at building a tree-like discourse structure of the text based on the linguistic expressions that state different kinds of relationships between sentences [12, 13]. This tree structure is used to define a salience function in order to select the more relevant sentences. Toshiba Company has developed a system based on this method. The system builds a rhetoric representation of the text that highlights all the rhetoric relationships between the sentences of the text (by using mainly linguistic markers such as connectors). Then, it computes the relationships between paragraphs of the text. This representation is then used to remove the relationships that are less relevant. For instance, in the sequence: "*sentence1. Thus sentence2*", *sentence2* is more relevant than the *sentence1* since *sentence2* abstracts *sentence1*. By performing such deletions of irrelevant sentences, the system produces an extract with the remaining relevant sentences. This method supposes the existence of linguistic connectors based on which the discourse structure is build; it is hence specific to particular genre of texts.

Linguistic Knowledge-based approach uses mainly linguistic words and phrases to extract sentences from a text [3, 14]. The approach analyzes a sentence in order to identify those words and phrases expressing its relevance according to the author's point of view. This approach is suitable for processing texts in various domains since words and phrases are not tied to a specific domain. SERAPHIN [2, 3] is a system developed according to this method. It uses a linguistic knowledge base and a set of rules that attribute labels to sentences. Then, sentences are extracted according to a label-based strategy and the length of the final abstract. In order to deliver a readable and coherent extract SERAPHIN uses a set of coherence rules which are used to solve references in the summary.

3. Summarization of Arabic texts

Summarizing Arabic texts in this paper consists in extracting the relevant sentences by using a variety of contextual information. In general, the context surrounding words is very significant when it comes to identify some semantic features of sentences. Indeed, the presence of specific linguistic expressions or words in a sentence, their order and relative locations convey the discursive intention of the author which is assigned a *semantic label*. For instance, linguistic expressions such as: "سنعرض في هذا المقال" ("*in this paper we present*") represents a thematic statement, "مهم جدا ملاحظة" ("*it is very important to notice*") represents a highlight statement and "أخيرا، خلاصة القول" ("*in sum, to conclude*") denotes a conclusion statement. These semantic labels assigned to sentences allow us to select the more relevant sentences.

In order to be able to analyze texts it is important to collect the linguistic expressions and to categorize them under semantic labels. This work is done during the knowledge acquisition phase which objective is to feed our system with the necessary linguistic knowledge to be used for summarization.

3.1. Linguistic Knowledge Acquisition and Modeling

A major obstacle facing the development of natural language engineering systems intended to deal with unrestricted text is the need for large amounts of linguistic knowledge to handle the complexity of language [15, 16]. By linguistic knowledge we mean the linguistic expertise that an expert linguist make use of to solve a problem related to the language. Depending on the nature of the problem to solve, linguistic knowledge needed may vary significantly from morpho-syntactic knowledge available in machine-readable dictionaries until deep knowledge describing subtle detail about semantic representation of words of the language. In order to encode linguistic knowledge into a program, we need to go through a crucial phase in knowledge engineering that is Knowledge acquisition. Knowledge acquisition aims at transferring the problem solving expertise from an expert or some knowledge source to a program. Generally the transfer is accomplished by a series of interviews between a domain expert: the linguist, and a knowledge engineer who then writes a computer program representing the knowledge.

Generally, knowledge acquisition is done through a four steps cycle *Elicitation, Representation, Implementation* and *Validation*. *Elicitation* consists of identifying and classifying the linguistic data and defining models that can map it. This step has been carried out mainly from a set of texts selected from online Arabic newspapers[1]. This process resulted in the identification of a set of linguistic patterns and the corresponding word lists. *Representation* consists of representing the knowledge in a formal language so that to be closer to the implementation. Then, the representation of the expertise must be turned into a runnable program. This is done in the *Implementation* step where the expertise has been expressed into the JAVA language. *Validation* is the last step where the expert has to test and verify the missing, incomplete or incorrect system data and rules.

3.2. Linguistic pattern acquisition

The expression of the Arabic linguistic expressions for summarization can extremely vary in terms of the grammatical structure and the vocabulary used. Fortunately, the number of the expressions frequently used to for each semantic label is relatively limited.

Semantic Label	Linguistic Pattern	Linguistic Expressions
Objective	LVNobjective + LNpaper LVNobjective + LNauthor	يقصد المقال يريد في مقاله يرمي الكاتب
Thematic	LVthematic + LNpaper LVthematic + LNauthor	يتطرق المقال هذا المقال يعرض يشرح الكاتب

Table 1 – Linguistic expressions and their corresponding patterns.

The first step towards defining linguistic patterns is to collect these expressions from corpus. This process has been done manually using a collection of texts. Linguistic expressions are then grouped under common linguistic patterns. Table 1 shows some of

[1] The texts are selected mainly from Al Ahram (http://www.ahram.org.eg/) and Al Hayat (http://www.daralhayat.com/) newspapers.

the linguistic expressions collected, their corresponding patterns and the associated semantic labels.

The right column of Table 1 shows the expressions corresponding to statements as found in corpus. The column in the middle includes the linguistic patterns that resulted from the aggregation of the expressions. The left column lists the semantic labels associated to the linguistic pattern.

3.3. Word list extension

Once the linguistic patterns and the word lists are defined, an important step is to extend the word lists so that to extend the language coverage of the linguistic patterns. Adding a word into a list must be done carefully. In general, synonyms are the first candidates to check. However, the set of synonyms as provided in dictionaries is too broad and encompasses the whole language. A fully automatic acquisition of synonyms from machine-readable dictionaries can give very odd results. This is why the extension needs to be controlled by two constraints. The first constraint is to consider the *synset* (synonym set); a set of words that are interchangeable in some contexts. This interesting feature is available in some machine-readable dictionaries[2]. The second constraint is to substitute in the original context the new word and then present it to the linguist who needs to validate the addition of the word in the targeted list. Therefore, word list extension is done mainly by *contextual synonymy* leaving the final decision to the expert.

List name	Common meaning	Word List
LVNobjective	القصد	قصد، رمى، أراد، هدف، ابتغى، أمل، غرَض، غاية، مراد، مقصد، ...
LNpaper	المقال	مقال، بَحْث، دِراسَة، مَبْحَث، ورقة، خَبْر، بيَان، تَصْرِيح،حديث، تَقْرِير، رِبُورْتاج، مَحْضَر، مَنْشُور، نَبَأ، نَشْرَة، ...
LVthematic	العرض	تطرق، شرح، عرض، بين، أبَان، أظْهَر، أنْهَج، أوْضَح، بَرَّز، بسَط، وَضَّح، بَسَط، سرد، قَدَّم
LNauthor	الكاتب	كاتب، باحث، محرر، ناقد، أستاذ، مُؤَلّف، مراسل،

Table 2 – Word list extension.

Table 2 shows the extended word lists corresponding to the linguistic patterns of Table 1. The first column is the list name as used in the system. The second column is the common meaning of the words included in the list. The third column presents the list of words.

[2] Sakhr's dictionary Al Qamoos (http://qamoos.sakhr.com) has a set of synonyms associated with practically each dictionary entry. Microsoft Word has a thesaurus including synonyms.

4. System Implementation

4.1. System Architecture

The summarization system architecture is presented in Figure 1. It includes three main modules. The *Text Processing* module is in charge of pre-processing the text in order to tokenize the text and split it into sentences. Then a stemmer reduces the words into their canonical form. The *Sentence Identification* module uses the linguistic database to identify all the words in the word-lists resulting from the knowledge acquisition phase. Then, the matching algorithm attempts to match the patterns with the sentence words. When a pattern matches an expression present in a sentence, the algorithm associates to the sentence the semantic label corresponding to the pattern. The *Sentence Selection* module selects the most relevant sentences of the text according to a compression factor defined by the user, and using a selection strategy that stipulates semantic label priority. At present, the strategies are not well defined, however we expect to define in the near future a couple of strategies representing different user needs.

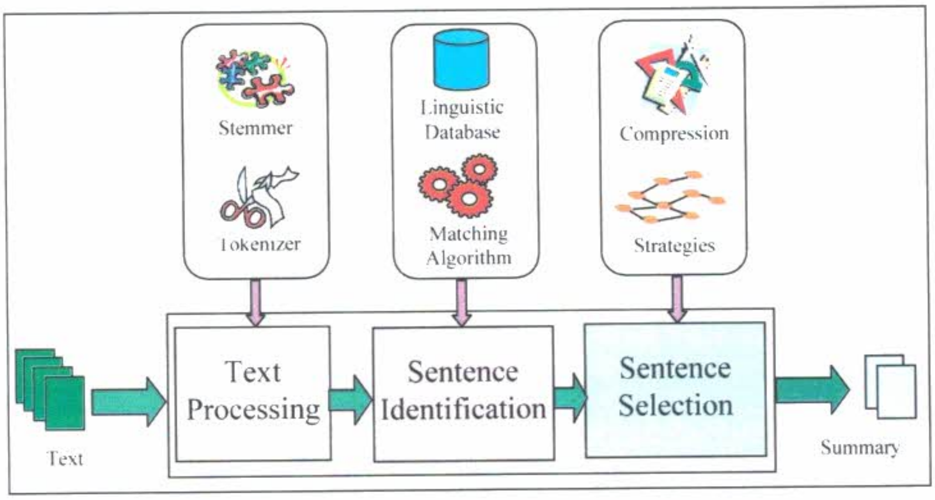


Figure 1 – System Architecture

4.2 Java based implementation

The system is implemented in JAVA programming language for many reasons i) it is platform independent, ii) it fully supports Unicode for Arabic characters, iii) it offers a large collection of ready made software components that provide useful graphical user interface capabilities, and iv) the possibility to run the system onto a browser.

The linguistic word lists are implemented as a database using *Cloudscape*[3] that is a pure, open source-based Java relational database management system which can be embedded in Java programs and used for online transaction processing. Cloudscape is a platform-

[3] Cloudscape™ V10.0 is a product of IBM. For more details see: www.cloudscape.com

independent database that integrates tightly with any Java-based solution. It can run on any standard Java Virtual Machine, allowing developers to "*write once, deploy anywhere*".

In order to create our database and insert the linguistic words, the following segment of code has been used

```
Class.forName("com.ihost.cs.jdbc.
        CloudscapeDriver").newInstance();
Connection con = DriverManager.getConnection("jdbc:cloudscape:
        ../database;create=true","root","secret");
Statement s = con.createStatement();
s.execute("create table linguisticWords
        (name VARCHAR(100), index INTEGER)");
String fixed [] = new String [3];
fixed = StreamConverter(3,"lang.txt");
  for(int i=0;i<3;i++)
  {
        System.out.println("fixed: "+fixed[i]);
        s.execute("insert into linguisticWords
              values ('"+fixed[i]+"', "+i+")");
  }
```

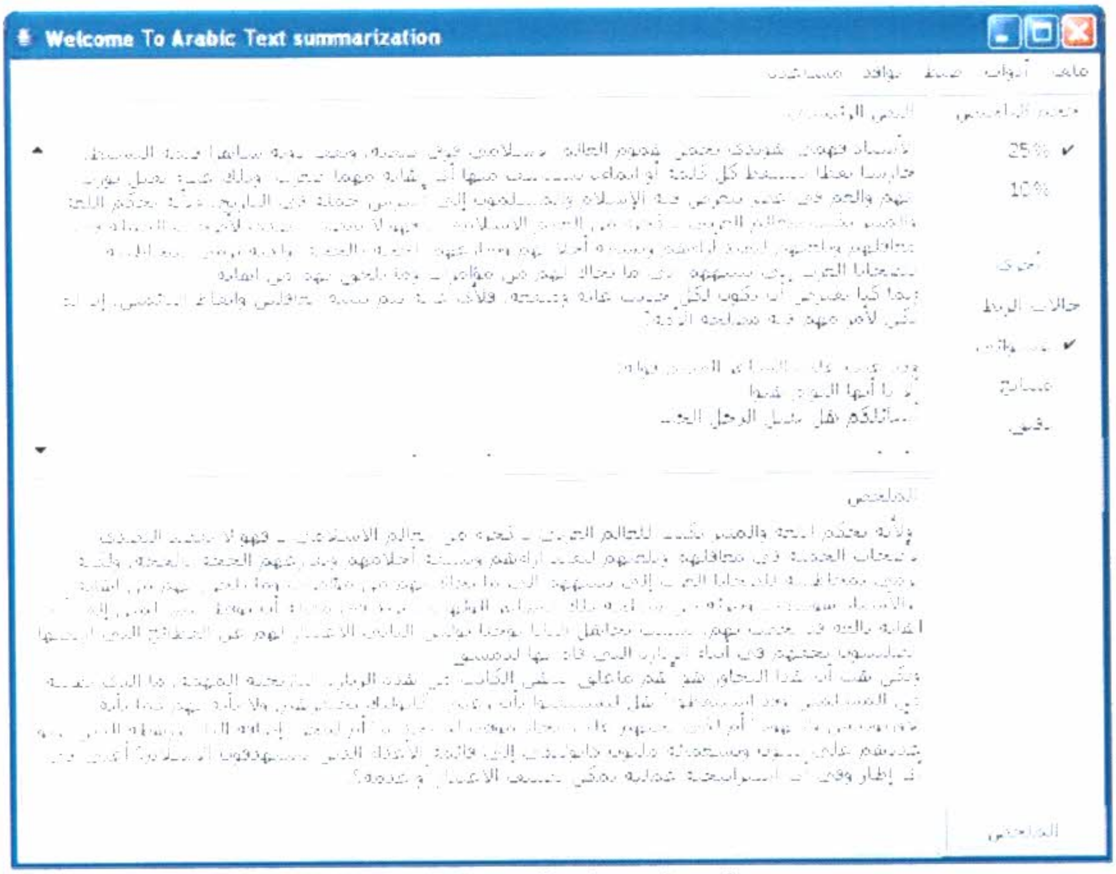
4.3. Matching Algorithm

The matching algorithm is a component that has already been used in other applications [17]. It matches linguistic patterns to sentences. It is completely decoupled from the data so that to allow the system to be updated easily. Hence, the linguistic patterns and the word lists can be updated without affecting the algorithm and vice-versa. The matching algorithm is implemented with three pattern matching options including the *Pure Sequential*, *Sequential* and *Random* search modes. These options were implemented in order to cope with the diversity of words to search for and to provide different search constraints which allow to loose or tighten the search depending on the texts to deal with. The *Pure Sequential* search mode represents the highest constrained search since it forces the system to match the words of the pattern with expressions in the text appearing in a strict consecutive order. Using this search mode no intermediate text tokens are allowed between the words of the patterns. The *Sequential* search mode requires the matched words to be in sequence but accepts the presence of alternate words inside the matched expression. The *Random* search mode necessitates the presence of all the words of the pattern in the sentence with no specific order.

Furthermore, the algorithm considers the length of the pattern (number of words in the pattern) as another constraint. Actually, the patterns are ordered according to their lengths and the algorithm starts with the most restrictive patterns that are patterns with the highest length. If no result is found, the algorithm considers the lower length patterns. This way of considering patterns allows the system to be more accurate in matching patterns and hence attributing usually the right semantic labels to sentences.

4.4. Graphical User Interface

Our system offers Arabic and English graphical user interfaces (GUI). Figure 2 shows the Arabic GUI version. The header of the main window presents the main menu which includes a set of facilities related to: i) the file to summarize, ii) tools related to the management of the word-lists in the database, iii) options for editing the original text and the summary, and finally iv) a help to ease the use of the system. In order to use the system, the user need to choose a text, specify the reduction ratio to apply to the original text and the matching option for matching linguistic patterns, these options are present in the right side of the main window. The original text[4] is displayed in the upper text window and the summary is displayed in the lower window.


Welcome To Arabic Text summarization
ملف أدوات ضبط نوافذ مساعدة
معدل التلخيص
النص الرئيسي
25% ✔
10%
أخرى
حالات الربط
الملخص

[illegible]

Fig 2 – The graphical user interface

[4] The original text in Figure 2 (upper text window) is an article from Al Ahram newspaper dated on 29th May 2001 written by Mohammed Ibrahim Al Chouch. The summary (in the lower window) represents 25% of the original text.

5. Conclusion

Summarization is gaining increasing interest while online information is overwhelming our everyday life. A summarizer main goal is to produce a condensed version of the content of an input text for various users. It is an elegant solution for users who are facing the thousands of web pages generally retrieved by the web search engines. The summarizer presented in this paper is able to summarize Arabic texts based on linguistic knowledge in the form of linguistic patterns and word lists representing language expressions. Language expressions express the language knowledge and are independent from any specific domain.

Our future work will be oriented toward two main directions: i) extend the linguistic knowledge base including linguistic knowledge to have reasonable language coverage, and ii) define a set of strategies to produce summaries based on user models.

6. References

[1] Desclés, J. –P.: *Langages applicatifs, Langues naturelles et Cognition*, Hermès, Paris, 1990.

[2] Berri J., Le Roux, D., Malrieu, D., Minel, J.-L., "SERAPHIN main sentences automatic extraction system", *proceedings of the Second Language Engineering Convention, London*, UK, 1995.

[3] J. Berri, *Contribution to the contextual exploration method. Applications to automatic abstracting and to temporal representations. Implementation of SERAPHIN system.*, Ph.D. thesis, Paris-Sorbonne University (Paris IV), France, 1996. (French publication)

[4] H. P. Luhn, "The automatic creation of literature abstracts", *IBM journal for research and development*, 2, 159-165, 1958.

[5] I. Mani, *Automatic summarization*, John Benjamins Publishing Company, Amsterdam, 2001.

[6] W. Kinsh, T. A. Van Dijk, "Toward a model of text comprehension and production", *Psychological review*, 85, 363-394, 1978.

[7] G. F. DeJong, "An overview of FRUMP system", *Strategies for natural language processing*, Lehnert and Ringle ed., Hillsdale, NJ: Erlbaum Ass., 149-172, 1982.

[8] R. Alterman and L. Bookman, "Some Computational Experiments in Summarization", *Discourse Processes*, 13, 143-174, 1990.

[9] G. Kallgreen, "Automatic abstracting of content in text", *Nordic journal of linguistics*, 11,89-110, 1988.

[10] G. SALTON, *Automatic Text Processing. The transformation. analysis, and retrieval of information computer*, Addison-Wesley, New York, 1989.

[11] P. Schäuble, "SPIDER : a multiuser information retrieval system for semistructured and dynamic data", *ACM SIGIR conference on R&D in information retrieval*, 318-327, 1993.

[12] S. Miike, E. Itoh, K. Ono, K. Sumita, "A full-text retrieval system with a dynamic abstract generation function", *Proceedings SIGIR'94*, ed. W. Bruce Croft and C. J. van Rijsbergen, Springer-Verlag, Dublin, 152-161, 1994.

[13] D. Marcu, "Discourse trees are good indicators of importance in texts", *advances in Automatic Text Summarization*, Ed. Inderjeet Mani and Mark T. Maybury, MIT press, USA, 123-136, 1999.

[14]C. D. Paice, "Constructing literature abstracts by computer: techniques and prospects", *Information processing management*, 26 (1), 171-186, 1990.

[15] Kim, J.-T., Moldovan I.: "Acquisition of Linguistic Patterns for Knowledge-Based Information Extraction" *IEEE Transactions on Knowledge and Data Engineering*, Vol. 7, No. 5, 713-724, Oct. 1995.

[16] Mädche, A., Neumann, G., Staab S.: 1999. "A Generic Architectural Framework for Text Knowledge Acquisition", *Unpublished Technical Report*, Kalsruhe University, 18p, 1999. Available at http://www.aifb.uni-kalsruhe.de/WBS

[17] J. Berri, M. Al-Khamis, "Information Exploration Using Mobile Agents", *WSEAS Transactions on Computers*, 3 (3), 706-712, 2004.