

An Ontology-based Summarization System for Arabic Documents (OSSAD)

Ibrahim Imam
Computer Science
Department, AASTMT
Al Mushir Ahmed
Ismail, Cairo, Egypt

Alaa Hamouda
Faculty of Computer
Engineering, Al Azhar
Nasr City, Cairo, Egypt

Hebat Allah Abdul Khalek
Computer Science Department,
AASTMT
Al Mushir Ahmed Ismail, Cairo,
Egypt

ABSTRACT

With the problem of increased web resources and the huge amount of information available, the necessity of having automatic summarization systems appeared. Since summarization is needed the most in the process of searching for information on the web, where the user aims at a certain domain of interest according to his query, domain-based summaries would serve the best. Despite the existence of plenty of research work in the domain-based summarization in English, there is lack of them in Arabic due to the shortage of existing knowledge bases. In this paper an Ontology-based Summarization System for Arabic Documents, OSSAD, is introduced. Domain knowledge is extracted from an Arabic corpus and represented by topic related concepts/keywords and the lexical relations among them. The user's query is first expanded by using the Arabic WordNet and then by adding the domain-specific knowledge base to the expansion. For summarization, decision tree algorithm (C4.5) is used, which was trained by a set of features extracted from the original documents. For the testing dataset, Essex Arabic Summaries Corpus (EASC) was used. Recall Oriented Understudy for Gisting Evaluation (ROUGE) was used to compare OSSAD summaries with the human summaries along with other automatic summarization systems, showing that the proposed approach demonstrated promising results.

General Terms

Automatic Summarization, Machine Learning, Ontology.

Keywords

Arabic text summarization, Knowledge-based summarization, Query expansion, Ontology extraction from text, Arabic WordNet.

1. INTRODUCTION

Due to increased access to data on the web, it is becoming harder to understand a certain topic without making the effort to read long documents and go through a lot of web pages to determine the most relevant ones. A need arose for automatic systems that would save the user's time, such as document clustering software, automatic summarizers, data mining software, etc. A summary can be defined as a text that is produced from one or more texts, that convey important information in the original text(s), and that is no longer than half of the original text(s) and usually significantly less than that. The main goal of a summary is to present the main ideas in a document in less space [1].

Many researches have been done in the field of automatic text summarization revealing different types of it. The most famous type is the extractive summarization, which takes as input a collection of text fragments (paragraphs or sentences) for one or more documents, and selects some subset into a summary. The fragments are then ranked, which allows

meeting different summary lengths requirements. The extraction techniques vary from using some important features to rank the fragments, to using a user's query to help indicating the fragments' importance, and to using a knowledge base to indicate the fragments' relevance to a specified domain.

When an online search engine is used, the user expects the results to be from a certain domain, as specified by the query. Therefore, query and domain knowledge-based summaries are needed. Several studies have showed that using a knowledge base in the process of extractive summarization improved results [2], they used knowledge from WordNet as well as from UMLS, a medical ontology, and this improved the performance. Some researchers preferred to manually build their knowledge by identifying a list of medical cue phrases and terms from a corpus of medical news articles, the sentences were ranked using important features plus adding the presence of domain specific phrases [3]. Others used encyclopedic knowledge in Wikipedia to expand user's query [4]. Some other researchers preferred to use ontologies to represent their knowledge, some use existing ontologies [5] and [6], where the user's query is expanded with synonyms and semantically related concepts using online available medical ontologies. And some other researchers constructed their ontology manually, where the authors manually construct an ontology for a small domain of news articles, using the category labels of the ontology tree to score paragraphs [7]. It is noted that most knowledge-based summarization systems don't use machine learning approaches to decide the most informative textual parts.

OSSAD automatically generates domain-related summaries. These summaries will be aimed at the user's interest and will take the user's query into consideration. The system automatically builds a specified domain ontology to produce a domain-related summary. As the ontology is built automatically, the user is free to choose the desired domain of interest, rather than being tied to a certain single domain. OSSAD uses the query as an input and expands it using the Arabic WordNet (AWN), which provides the system with the words' lexical synonyms. It then produces a knowledge base that consists of the domain ontology concepts and relations. The user's query is expanded with the knowledge base, and the sentences are given scores (features) according to their relevance to the original and expanded versions of the query. C4.5 decision tree algorithm is trained using the set of sentences' extracted features. Finally, when the C4.5 trained model is applied to the testing data, sentences of the final summary are displayed to the user. The knowledge base and training corpuses were collected from the World Wide Web. The Essex Arabic Summaries Corpus (EASC) was used for the testing data because it has many articles grouped by topic,

with multiple human-generated summaries for each article. The chosen topic was the “Environment” domain, as it had the largest number of related articles.

This paper is organized as follows: Section 1 is the introduction. Section 2 is an overview of the system components. Section 3 is the description of the details of the proposed system. Then section 4 comes to evaluate the proposed system. And finally section 6 is the conclusion and future work.

2. SYSTEM COMPONENTS OVERVIEW

Before describing the details of the proposed system; OSSAD, an overview of the system components is illustrated in this section.

2.1 Ontology

The ontology is an explicit, formal specification of a shared conceptualization of a domain of interest. It should be restricted to a given domain of interest and therefore model concepts and relations that are relevant to a particular task or application domain [8]. Ontologies provide a richer knowledge representation that improves machine interpretation of data [9].

Manual acquisition of ontologies is a tedious and cumbersome task. It requires an extended knowledge of a domain and in most cases the result could be incomplete or inaccurate. Manually built ontologies are expensive, error-prone, and biased towards their developer. Researchers try to overcome these disadvantages of manual building of ontologies by using semi-automatic or automatic methods for building ontologies. Automation of ontology construction not only reduces costs, but also results in an ontology that better matches its application. During the last decade, several ontology learning approaches and systems have been proposed. They try to build ontology by two ways. One way is developing tools that are used by knowledge engineers or domain experts to build the ontology like Protégé and Jena. They are called the ontology modeling tools. Another way is semi-automatic or automatic building of ontologies by learning it from different information sources [10]. The next two sub-sections talk about the two methodologies for automatic or semi-automatic ontology building.

2.2. Ontology Learning From Text

Ontology learning refers to extracting ontological elements (conceptual knowledge) from input and building ontology from them. It aims at semi-automatically or automatically building ontologies from a given text corpus with a limited human expert. The ontology building can be from scratch (automatic), or by adapting an existing ontology in a semi-automatic fashion using several sources [10].

Text or unstructured data is the most difficult type to learn from. It needs more processing than the semi-structured or structured data. The systems which have been proposed for learning from free text consists of the following four main processes:

2.2.1 NLP

An ontology extractor from text must perform some NLP processes on the corpus. In a matter of fact some pre-processing should be applied on texts before NLP is, such as removing abbreviations, numbers, words that don't belong to the ontology language, diacritics (تشكيل) in case of Arabic, etc. NLP processes include POS taggers, parsers (shallow or dependency), NER (Named Entity Recognizer), removing stop words, or stemming/lemmatizing.

2.2.2 Concept Extraction

Concept or keyword extraction can be described as the task of identifying a small set of words, key phrases, keywords, or key segments from a document that can describe the meaning of the document [11]. The existing methods for keyword extraction were divided into four categories [11], (a) simple statistics such as term frequency. (b) Linguistic analysis such as POS tagging, the analysis are mostly combined with statistical measures. (c) Machine learning.

2.2.3 Relation Extraction

Ontologies, besides having a list of concepts, should define the relations among concepts. Several researchers have attempted to find taxonomic relations expressed explicitly in texts by matching certain patterns which is referred to as Hearst-patterns. Other researchers have used the internal structure of noun phrases. Some other relations are determined using Harris's distributional hypothesis [12].

2.2.4 Ontology or Hierarchy Building

Some of the ontology extraction tools have reference ontology to update it with the new concepts and relations deduced by the tool. Other tools use agglomerative, divisive or incremental clustering to build the ontology hierarchy. And finally some tools use Formal Concept Analysis (FCA), which is a principled way of deriving a *concept hierarchy* or formal ontology from a collection of objects and their properties [13]. There are some available tools that extract ontology from text, such as Text-To-Onto, and its successor text2Onto, OntoLearn, protégé plugin OntoLT, TERMINAE and some other done by researchers such as CRCTOL and automatic construction of ontology from Arabic texts [20].

2.3 AWN (Arabic WordNet)

WordNet is a lexical database for the English language. It groups English words into sets of synonyms called *synsets*, providing short, general definitions, and recording the various semantic relations between these synonym sets. The purpose is to produce a combination of dictionary and thesaurus that is more usable, and to support automatic text analysis and artificial intelligence applications. Though WordNet contains a sufficiently wide range of common words, it does not cover special domain vocabulary, since it is primarily designed to act as an underlying database for different applications [14]. The AWN follows the methodology of the EuroWordNet [15].

3. SYSTEM DESCRIPTION

The system includes three main components, the first component is responsible for the knowledge construction, the second one does the sentence feature extraction, and finally summarizing using decision trees. Fig. 1 shows OSSAD architecture.

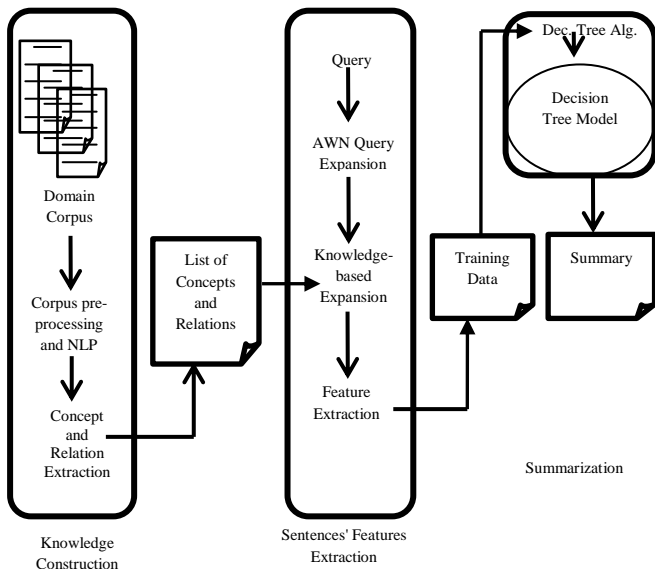


Fig 1: OSSAD architecture

3.1 Knowledge construction

3.1.1 Corpus Pre-Processing and NLP

More than one hundred Arabic articles were collected from the internet. Some pre-processing is made to the corpus before knowledge extraction can be applied. Any non-Arabic words or letters, numbers, diacritics (تشكيل), symbols or non- letters such as brackets or quotations, extra spaces or empty lines are removed.

Stanford POS tagger is then used to determine the type of each word, i.e. noun, verb, etc. Stanford POS tagger is a Java implementation of the log-linear part-of-speech taggers [16].

3.1.2 Concept Extraction

In OSSAD, multi-word concepts were extracted, which are composed of words that co-occur together more often than can be expected by chance. According to some studies [17], most domain-specific concepts are multiword terms. The small number of relevant single-word terms can either be found appearing frequently in the multiword terms or easily inferred based on the multiword terms. Also single word concepts may include general concepts as well as domain ones, and the relation extraction might not be easy especially in the Arabic language, with the lack of dependency parsers and human intervention.

Some of the approaches used in literature to extract multi-word concepts, [18], use statistical approaches then apply frames on the results to exclude the unwanted patterns [20], [21], and some other approaches apply parsing or use POS tagger to choose certain patterns such as Text-To-Onto. All approaches then calculate the frequency of each pattern among the corpus, some of them stop at that step and some others extend the calculations to enhance performance such as using Domain Relevant Measures (DRM), Terminology Identification Measure (TIM) [17], or the C-value method. In OSSAD, the C-value measures were used because in the previous researches it showed better precision and recall than to just depend on the frequencies [18]. The C-value method is an efficient domain-independent multi-word term recognition method which combines linguistic and statistical knowledge [19].

The C-value algorithm [22] includes the following steps:

- 1- Tag the corpus.
- 2- Choose certain patterns for candidate concepts like *Noun+Noun*, *(Noun/Adj)+Noun*, *(Noun/Adj)+(Adj/Noun)*. In the algorithm the maximum number of words to form a concept was not defined. But in Arabic language it is reasonable not to consider more than four words [21], [20].
- 3- Calculate the total frequency of the candidate string in the corpus.
- 4- Calculate the frequency of the candidate string as part of other longer candidate terms.
- 5- Calculate the number of these longer candidate terms.
- 6- Calculate the length of candidate strings (in number of words).
- 7- Calculate the C-value.

Figure 2 presents an example of the concepts extracted for the environment domain:

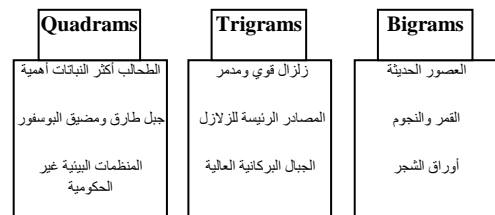


Fig 1: Concepts

3.1.3 Relation extraction

Techniques for taxonomic relation extraction vary from using a dependency parser to determine the relation among concepts to statistical approaches. Some approaches use dependency parsers to determine if one concept is the subject of another, which indicates a relation, while other tools use shallow parsers or POS taggers and consider the pattern (Noun₁, Verb, Noun₂) [17] depending on that verbs are hypothesized to indicate semantic relations between concepts. Some other approaches consider the relations of "is-a" and "has-a" [20], and some approaches use a mixture of that. All these approaches also detect the frequency of concepts that appear together. Text2Onto [23], developed JAPE patterns for both shallow parsing and the identification of concepts and different types of relations. JAPE rules have to be developed by humans who are aware of the domain, and the rules are processed using GATE; the NLP tool. Text2Onto supports only the English language for the easiness of JAPE rules creation.

Due to the lack of Arabic dependency parsers, in OSSAD, relations detected by Aliane [20] were adopted. They include "تتألف من... تتكون من", and "has-a" relations such as "هو... هي... هما...". Harris's distributional hypothesis for verbs was also adopted. Some other rules were figured out by conducting an analysis of the Arabic language nature, such as: resemblance of two concepts "مثل", and the ownership of a concept over the other "له... لها". The list of relations is prone to stretching if any new phrases were discovered. Pairs of concepts that meet one another more frequently than can be expected by chance are considered in a relation.

In OSSAD, the ontology hierarchy wasn't built, because it is out of the scope of the summarization and the hierarchy wouldn't add a value to the already built knowledge of concepts and relations.

3.2 Sentence Feature Extraction

3.2.1 AWN Query Expansion

The user's query is expanded by running it against the Arabic WordNet. AWN doesn't provide JAVA API like the EuroWordNet, so the AWN database along with the source code had been used to retrieve all the synsets. A list of all the expanded synonyms for all the words in the original query is checked to remove any lexical redundancies. The following is an example of a query (البيئة) and the result of its expansion: بيئة, محيط, وسط.

3.2.2 Knowledge Base Query Expansion

The original query is expanded against the knowledge base of concepts and relations. All the related concepts to every word in the query and their relations are added.

3.2.3 Feature extraction

A set of training data was produced to train the decision tree algorithm. An Arabic corpus of 40 articles was collected from the internet in the environment domain. Original articles are divided into sentences, based on the period punctuation symbol. All sentences are pre-processed to remove noisy text. Khoja stemmer is used to stem the sentences, the original query, the AWN expanded query and the knowledge-based expanded query. Every sentence is given a set of feature scores, the first score presents the number of occurrences of words from the original query in the sentence, the second is for the number of occurrences of words from the AWN expanded query, the third is for the concept expanded query and the fourth is for the relation expanded one. Every sentence is represented in the form of a vector of its features, containing the sentence along with its four features scores. Every vector is given one of the following three labels, most important (informative) and domain-related represented by (3), domain-semi-related or less important represented by (2) and domain-unrelated and less important represented by (1). These scores indicate the degree of a given sentence's relation to the domain of interest (*environment*) and its importance (*informativeness*). Each sentence is revised in the different human summaries to give a score.

After that, the scores of human experts are averaged to express the final rank for each sentence.

3.3 Summarization

3.3.1 Decision trees algorithm

C4.5 builds decision trees from a set of training data, where the training data is made up of already classified samples. After C4.5, C5.0 was devised for Unix/Linux and See5 for Windows. C5.0 uses additional data types including dates, times and timestamps, and the Windows version has a user-friendly graphical interface. C4.5 has a free source code and the multiple programs used for generating decision trees in C4.5 have been merged into a single program in C5.0. While C5.0 has some features that are more advanced than those in C4.5, these updates are inconsequential to the results shown in this paper. Also C4.5 is a free and downloadable which is not the case for the See5.0.

3.3.2 Testing data

Essex Arabic Summarization Corpus (EASC) was used. It was created by Mahmoud El-Haj to address the shortage of relevant data for Arabic natural language processing [31]. Each text file is split into sentences, where unlabeled sentences are represented by a feature vector. The model created for the C4.5 algorithm is applied to each group of sentences representing a single file. According to the required

summary length, the appropriate number of sentences is displayed to the user.

4. EVALUATION AND RESULTS

Evaluation of summarization is a quite hard problem. Often, a lot of manual labor is required, for instance by having humans read generated summaries and grading the quality of the summaries with regards to different aspects such as information content and text clarity. Manual labor is time consuming and expensive. Summarization is also subjective. The conception of what constitutes a good summary varies a lot between individuals, and of course also depending on the purpose of the summary [30].

Some automatic text summarization tools use Document Understanding Conference (DUC) datasets to test their algorithms and some of them use human evaluation such as [24], while others use the abstract of an article as the human summary [6].

According to the lack of Arabic datasets or proper Arabic papers with abstracts, in OSSAD, EASC was used. It contains 153 Arabic articles and 765 human-generated extractive summaries of those articles. These summaries were generated using Mechanical Turk [26], which is a subsidiary of Amazon.com that provides a Web services system that uses people to perform tasks better handled by humans than computers [27]. This data also suited domain knowledge-based summarization; because the articles are divided into topics, i.e. art & music, education, environment, finance, health, politics, religion, science and technology, sports, and tourism. The environment domain was used in testing. Other human-generated summaries were also used for testing. Three people were enlisted to generate the summaries; two are language and literature experts, and the third is a computer science graduate. Each one was given the set of *environment* articles and asked to produce a summary for each article by extracting the most suitable sentences. All three human-generated summaries were 40% of the length of the original document.

The OSSAD summary was also evaluated against other Arabic summarization systems working on the same data. To achieve this, two summarization outputs from two systems were used, both developed by El-Haj, working on EASC corpus. Gen-Summ is a query-based document summarizer based on the vector space model. And LSA-Summ is similar to Gen-summ, but where the vector space is transformed and reduced by applying Latent Semantic Analysis (LSA) to both document and query [31].

The latest version of ROUGE was the method used to evaluate the OSSAD output. ROUGE-L only was considered in results' analysis, as ROUGE-L builds its comparison on the sentence level, which is the unit used by the OSSAD extraction system. ROUGE-1 also obtained the same scores as ROUGE-L.

Table 1. COMPARION between OSSAD and other summarization systems, and between OSSAD and human experts

	Average Recall	Average Precision
Human1 with Human2	56%	52%
Gen-Summ with (Human1 and Human2)	46%	39%
LSA-Summ with (Human1 and Human2)	49%	41%
OSSAD with (Human1 and Human2)	53%	47%

The results of the summary comparison are shown in table I showing the average precision and recall scores for comparing the three summarization systems with two gold standard summaries at a time. The purpose of comparing a sample of the human summaries was to show the degree of variance among them, and to give a measure for the best possible difference between gold standard summaries. The table also shows that in both the average precision and the average recall, OSSAD had better scores than the other two systems. The precision and recall for human1 with human2 shows to which extent the summarization process is subjective. There is a notable difference between the human summarization. OSSAD could reach a good level of performance (precision of 53% and recall of 47%) compared to the humans performance (precision of 56% and recall of 52%). It is important to note that the length of the original files to be summarized is low. It is about 10 sentences on average. This nature of data decreases the overall performance for all systems working on this data set as well as the human experts themselves.

5. CONCLUSION AND FUTURE WORK

OSSAD solves the problem of having to sort through numerous lengthy documents that are produced as the results of an online search by providing shorter, informative versions of the original documents. It is a user-focused summarization system, as it takes the query into consideration. And because all the results are expected to be from the same domain, OSSAD also uses domain knowledge to enrich the summary with domain related information. Unlike other knowledge-based summarization systems that use an existing ontology of a certain domain, OSSAD automatically extracts the domain ontology from any given corpus, which offers a domain freedom just by having any corpus of text files that belongs to the same domain. Finally, the information to be displayed in the final summary is selected using a decision tree algorithm and not just by taking the highest ranked information. The results show that OSSAD performance overcomes other systems and reach a good level of performance (precision of 53% and recall of 47%) compared to the humans (precision of 56% and recall of 52%).

The possible future work for this paper includes considering a confidence measure for the relationships extracted from concept pairs, extracting relationships at a higher level of abstraction and using a validation source for the set of relations extracted, such as the AWN or the WWW.

6. REFERENCES

- [1] Dragomir R. Radev, Kathleen McKeown, “*Introduction to the Special Issue on Summarization*”, Computational Linguistics – Summarization, Vol 28, No. 4, pp. 399-408, 2002.
- [2] Rakesh Verma, Ping Chen, Wei Lu, “*A Semantic Free-text Summarization System Using Ontology Knowledge*”, IEEE Transactions on Information Technology in Biomedicine, Vol 5, No. 4, pp. 261-270, 2007.
- [3] Kamal Sarkar, “*Using Domain Knowledge for Text Summarization in Medical Domain*”, International Journal of Recent Trends in Engineering, Vol 1, No. 1, pp. 200-205, 2009.
- [4] Vivi Nastase, “*Topic-Driven Multi-Document Summarization with Encyclopedic Knowledge and Spreading Activation*”, conference on Empirical Methods in Natural Language Processing, Waikiki, Honolulu, Hawaii, 2008.
- [5] A. A.Kogilavani, B. Dr.P.Balasubramanie, “*Ontology Enhanced Clustering Based Summarization of Medical Documents*”, International Journal of Recent Trends in Engineering, Vol 1, No. 1, pp. 546-549, 2009.
- [6] Ping Chen, Rakesh Verma, “*A Query-based Medical Information Summarization System Using Ontology Knowledge*”, Computer-based Medical Systems (CBMS), 19th IEEE International Symposium, USA, pp. 37 – 42, 2006.
- [7] Chia-Wei Wu, Chao-Lin Liu, “*Ontology-based Text Summarization for Business News Articles*”, ISCA 18th International Conference on Computers and Their Applications, Honolulu, Hawaii, USA, pp. 389-392, 2003.
- [8] Paul Buitelaar, Philipp Cimiano, Bernardo Magnini, *Ontology Learning from Text: Methods, Application and Evaluation*, IOS Press, 2003.
- [9] Ivan Bedini, Benjamin Nguyen, “*Automatic Ontology Generation: State of the Art*”, Molecular Evolution, Vol 44, No. 2, pp. 226-233, 1997.
- [10] Maryam Hazman, Samhaa R El-Beltagy, Ahmed Rafea, “*A Survey of Ontology Learning Approaches*”, Vol 22, No. 9, pp. 36-43, 2011.
- [11] Elena Demidova, Iryna Oelze, “*Automatic Keyword Extraction for Database Search*”, PhD thesis, University of Hannover, 2009.
- [12] Philipp Cimiano, Aleksander Pivk, Lars Schmidt-Thieme, Steffen Staab, “*Learning Taxonomic Relations from Heterogeneous Evidence*”, In: *Ontology Learning from Text: Methods, Applications and Evaluation*, pp. 59-73, IOS Press, 2005.
- [13] Wikipedia, http://en.wikipedia.org/wiki/Formal_concept_analysis, (10-01-2013).
- [14] Wikipedia, <http://en.wikipedia.org/wiki/WordNet>, (10-01-2013).
- [15] William BLACK, Sabri ELKATEB, “*Introducing the Arabic WordNet Project*”, Third International WordNet Conference (GWC-06), Korea, 2006.

- [16] The Stanford Natural Language Processing Group, <http://nlp.stanford.edu/software/tagger.shtml>, (14-01-2013).
- [17] Xing Jiang, Ah-Hwee Tan, “*Mining Ontological Knowledge from Domain-Specific Text Documents*”, Data Mining, Fifth IEEE International Conference, Singapore, 2005.
- [18] Euthymios Drymonas, “*Exploring multi-word similarity measures for Information Retrieval applications: the T-SRM method*”, PhD thesis, Technical University of Crete (TUC), Department of Electronics and Computer Engineering, 2006.
- [19] Sophia Ananiadou, Hideki Mima, “*An Application and Evaluation of the C/NC-value Approach for the Automatic term Recognition of Multi-Word units in Japanese*”, International Journal of Terminology, Vol 6, No. 2, pp. 175–194, 2000.
- [20] Ahmed Cherif Mazari, Hassina Aliane, Zaia Alimazighi. “*Automatic construction of ontology from Arabic texts*”, ICWIT, Vol 867, pp. 193-202. 2012.
- [21] Mohammed Attia, Antonio Toral, Lamia Tounsi, Pavel Pecina, “*Automatic Extraction of Arabic Multiword Expressions*”, the 7th Conference on Language Resources and Evaluation (LREC), 2010.
- [22] Katerina Frantzi, Sophia Ananiadou, Hideki Mima, “*Automatic recognition of multi-word terms: the C-value/NC-value method*”, International Journal on Digital Libraries, Vol. 3, No. 2, pp. 115-130, 2000.
- [23] Philipp Cimiano, Johanna Völker, “*Text2Onto - A Framework for Ontology Learning and Data-driven Change Discovery*”, 10th International Conference on Applications of Natural Language to Information Systems (NLDB), Spain, pp. 227-238, 2005.
- [24] Mahmoud O. EL-HAJ, Bassam H. HAMMO, “*Evaluation of Query-Based Arabic Text Summarization System*”, Natural Language Processing and Knowledge engineering International Conference, IEEE, Jordan, pp. 1-7, 2008.
- [25] Mahmoud El-Haj, Udo Kruschwitz, Chris Fox, “*Multi-Document Arabic Text Summarization*”, Computer Science and Electronic Engineering Conference (CEEC), IEEE, UK, pp. 40 – 44, 2011.
- [26] Summarisation Corpora, <http://privatewww.essex.ac.uk/~melhaj/easc.htm>, (14-01-2013).
- [27] PCMAG.com, http://www.pcmag.com/encyclopedia_term/0,1237,t=Mechanical+Turk&i=57289,00.asp, (14-01-2013).
- [28] ROUGE, <http://www.berouge.com/Pages/DownloadROUGE.aspx>, (14-01-2013).
- [29] Kavita Ganesan, ChengXiang Zhai, Jiawei Han, “*Opinosis: A Graph-Based Approach to Abstractive Summarization of Highly Redundant Opinions*”, the 23rd International Conference on Computational Linguistics (COLING '10), China, 2010.
- [30] Jonas Sjöbergh, “*Older versions of the ROUGEeval summarization evaluation system were easier to fool*”, the International Journal of Information Processing and Management, Vol. 43, No. 6, pp. 1500-1505, 2007.
- [31] Mahmoud El-Haj, Udo Kruschwitz, Chris Fox, “*Using Mechanical Turk to Create a Corpus of Arabic Summaries*”, the Seventh conference on International Language Resources and Evaluation, Valletta, Malta, 2010.