Point and Paste Question Answering

Sanda M. Harabagiu
Department of Computer Science
University of Texas at Dallas
Richardson, Texas, 75083-0688
sanda@cs.utdallas.edu

Finley Lăcătuşu
Department of Computer Science
University of Texas at Dallas
Richardson, Texas, 75083-0688
finley@student.utdallas.edu

Paul Morărescu
Department of Computer Science
University of Texas at Dallas
Richardson, Texas, 75083-0688
paulm@student.utdallas.edu

Abstract

The informative answer to a question needs sometimes to be formatted as a summary of a document or set of documents. This scenario requires a novel Question Answering architecture that relies on a combination of topical relations mined from WordNet and textual information available from documents related to the question's topic. This paper presents a methodology of pointing out information to be included in a summary by using extraction templates. The paper also describes how the summary is generated using a small set of pasting operators.

Introduction

Automatic textual Question Answering (QA) draws substantial interest both from research laboratories and the commercial world because it provides a promising solution to the information overload we are currently facing. Instead of browsing through long lists of documents whenever we need answers to a question, we rather read a short text snippet containing the answer we are looking for. One interesting aspect of QA is uncovered if we take into account the observation that not all topics are created equal. In a given on-line text collection, some topics may be covered by more documents whereas other topics are briefly mentioned. When a question asks about one of the more popular topics, summarization techniques need to replace that standard paragraph retrieval techniques used in current QA systems. If a document covers at length one of the topics, any of the paragraphs is relevant to the question asking about that topic. Instead of selecting one of the paragraphs based on some keyword-based metric, we argue that using summarization techniques will produce more informative answers. If several documents cover different aspects of the same topic, the answer should be generated by multi-document summarization techniques similar to those presented in (Radev and McKeown 1998).

In this paper we revisit the problem of QA by focusing on questions that are answered by a summary. We thus study the changes imposed on the three typical processes of a QA system: (1) question processing; (2) document processing and (3) answer formulation. Moreover, the techniques we employ for generating summaries as answer to question are different from those developed for query-based summarization. At the core of our method of generating topic-related summaries is the lexico-semantic knowledge that we discern for each topic in part. To be able to identify textual information that is relevant to a question we reply on pattern matching of linguistic rules against free text. Similar to the Information Extraction (IE) task developed for the Message Understanding Conferences (MUC), if a topic template is known, we can rely on linguistic rules that recognize the topic information. However, when a question is asked, no template is provided. Instead, we generate an ad-hoc template based on topical relations that are mimed from the WordNet lexico-semantic data. When the topic is covered by a single document, we generate a summary using sentences that are extracted and compresses based on lexico-semantic and cohesive information brought forward by the WordNet topical relations. If the topic is covered by multiple documents, the summery is generated by merging topic templates and using some of the possible relations between templates. This form of fusing information is highly dependent on the combination of topic patterns mined from WordNet and the lexico-semantic relations discovered in texts.

The remaining of this paper is organized as follows. Section 2 presents the topical relations that can be mined from WordNet, whereas Section 3 presents the techniques for generating point-and-paste answers. Section 4 summarizes the conclusions.

Topical Relations

Lexical chains are sequences of semantically related words found in a document or across documents. They have been used extensively in computational linguistics to study: discourse, coherence, inference, implicatures, malapropisms, automatic creation of hypertext links, and others (Morris and Hirst 1991), (Hirst and St-Onge 1998), (Green 1999), (Harabagiu and Moldovan 1998). Current lexical chainers take advantage only of the WordNet relations between synsets, totally ignoring the glosses. A far better lexical chaining technology can be developed based on Extended WordNet since the interconnectivity of synsets will largely increase via the gloss concepts. Consider the following text: *The mine has revitalised diamond production.* One cannot

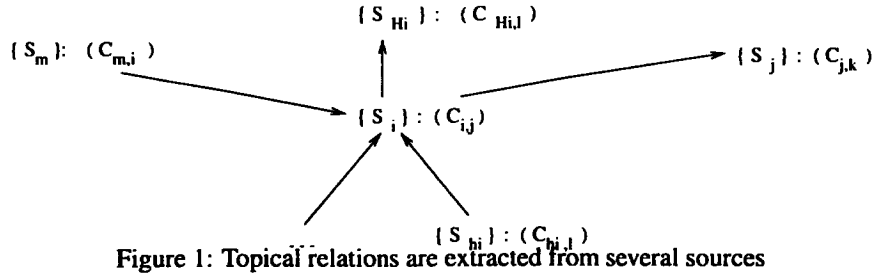

Figure 1: Topical relations are extracted from several sources

Synset	Gloss	Morphological relation
laughter:n#1	(the sound of laughing:v#1)	noun - verb
immediately:r#3	(bearing an immediate:a#2 relation)	adverb - adjective
insure:v#4	(take out insurance:n#3 for)	verb - noun
parental:adj#2	(..characteristic of or befitting a parent:n#1)	adjective - noun

Table 1: Examples of new morphological relations revealed by the topical relations

explain the cohesion, and the intention of this simple text by using only the existing lexico-semantic relations like those encoded in WordNet (Fellbaum 1998). Typical relations encoded in lexico-semantic knowledge bases like WordNet comprise *hypernymy*, or IS-A relations, *meronymy* or PART-OF relations, *entailment* or *causation* relations. None of these relations can be established between any of the words from the above text. However, it is obvious that this short text is both cohesive, as words have more than syntactic relations joining them, and coherent, as it is logical. Relations that account for the cohesion of the above text can be mined from WordNet's glosses that define the encoded concepts. In WordNet each concept is represented through a synonym set of words sharing the same part-of-speech. Such synonym sets are known ans *synsets*. The words used to define the synset {*mine*} are related to the words used to define the synset {*diamond*}. Furthermore, these two synsets are related to other synsets containing words such as *ring*, *bracelet* or *gem* and *precious*. Similarly, *production*, *market* and *mine* are related to the name of big diamond concerns like *De Beers*. All these relations converge toward the topic of *"diamond market"*, bringing forward all the various concepts that define the topic. This is why we call them *topical relations*.

A good source for topical relations is WordNet, namely the glosses defining each synset. Topical relations are pointers that link a synset to other related synsets that may occur in a discourse. Topic changes from a discourse to another, and the same concept may be used in different contexts. When the context changes, the concepts used change. For example, the diamond market can be discussed from a financial, or geological, or gem-jewelry point of view. In each case, the concepts used would be quite different. This pluralism of topics for the same concept creates confusion and difficulty in identifying related concepts.

The sources for the topical relations are illustrated in Figure 1.

1. The first place where we look is the gloss of each concept. Since the gloss concepts are used to define a synset, they clearly are related to that synset. From the gloss definition we extract all the nouns, verbs, adjectives and adverbs, less some idiomatic expressions and general concepts that are not specific to that synset. For example, the gloss of the concept *diamond* is *(a transparent piece of diamond that has been cut and polished and is valued as a precious gem)*

2. Each concept $C_{i,j}$ in the gloss of synset S_i points to a synset S_j that has its own gloss definition. The concepts $C_{j,k}$ may also be relevant to the original synset S_i. Although this mechanism can be extended further, we will stop at this level. For the case of the gloss of *diamond*, we shall have pointers to the glosses of *gem*, *cut*, *polish*, *valued* and *precious*.

3. A third source is the hypernym S_{Hi} and its gloss with concepts $C_{Hi,l}$. For example the synset {*jewelry, jewelry*} is one of the hypernyms of *diamond*, having its own gloss. Other relations such as causation, entailment and meronymy are treated in the same way as the hypernymy.

4. Yet another source consists of the glosses in which synset S_i is used. Those synsets, denoted in the figure with S_m are likely to be related to S_i since this is part of their definitions.

This scheme of constructing topical relations connects a synset to many other related synsets, some from the same part of speech hierarchy, but many from other part of speech hierarchies. The increased connectivity between hierarchies is an important addition to WordNet. Moreover, the creation of topical relations is a partial solution for the derivational morphology problem that is quite severe in WordNet. In Table 1 we show a few examples of morphologically related words that appear in synsets and their glosses which will be linked by the new topical relations.

Paths between synsets via topical relations

Figure 2 shows five synsets S_1 to S_5, each with its topical relation list represented as a vertical line. In the list of synset S_i there are relations r_{ij} pointing to S_j.

It is possible to establish some connections between synsets via topical relations. We propose to develop software that automatically provides connecting paths between any two synsets S_i and S_j up to a certain distance as shown

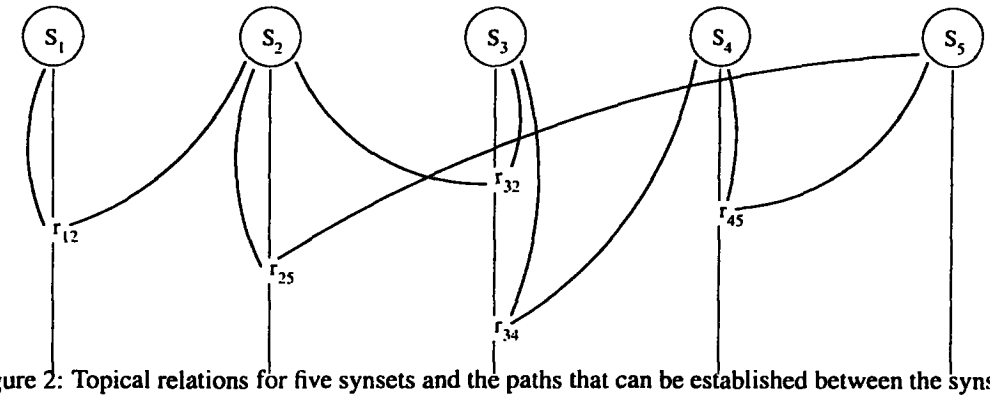

Figure 2: Topical relations for five synsets and the paths that can be established between the synsets

in Table 2. The meaning of these paths is that the concepts along a path have in common that they may occur in a discourse, thus are topically related. These paths should not be confused with semantic distance or semantic similarity paths that can also be established on knowledge bases by using different techniques.

Name	Path
V - path	$S_i - S_j$
W - path	$S_i - S_k - S_j$
VW - path	$S_i - S_k - S_l - S_j$
WW - path	$S_i - S_k - S_l - S_m - S_j$

Table 2: Types of topically related paths

The name of the paths was inspired by their shape in Figure 2. The V path links directly S_i and S_j either via a $r_{i,j}$ or $r_{j,i}$, namely either S_i has S_j in its list of topical relations or vice versa. The W path allows one other intermediate synset between S_i and S_j. Similarly, the VW path allows two intermediate synsets and the WW path allows three intermediate synsets. We limit the length of the path to five synsets. For each type of path, there are several possible connections depending on the direction of the connection between two adjacent synsets (i.e. via $r_{i,j}$ or $r_{j,i}$). These connections are easily established with known search methods. When searching for all topical relations of a given concept, we may assume that we try to establish whether there are topical relations between that concept and any other concept encoded in WordNet. To speed-up the search, we rely on hash tables of all possible paths between any sysnet and any other possible sysntet from WordNet. In this way, at processing time, retrieving all concepts that characterize a topic is a fast and robust process.

Examples

Suppose one wants to find out whether or not there are any topical relations between two words. The system will try to establish topical paths between any combination of the senses of the two words. Here are some examples:

Example 1: Is there any topical relation between *diamond* and *polish*?
Answer: there is a V-path: *diamond*:n#1 → *polish*:v#1 [1]
Explanation: *polish*:v#1 is part of the definition of synset *diamond*:n#1, thus it is on its topical relations list.

Example 2: Is there any topical relation between *ring* and *diamond*?
Answer: there is a W-path: *diamond*:n#1 → *gem*:n#1 ← *ring*:n#1
Explanation: *gem*:n#1 is found in the definition of *diamond*:n#1 and is also found in the definition of `ring`:n#1.

Example 3: Is there a topical connection between *diamond* and *gold*?
Answer: there is a VW-path: *diamond*:n#1 → *gem*:n#1 ← *ring*:n#1 → *gold*:n#1
Explanation: *diamond*:n#1 has *gem*:n#1 in its definition, which in turn is in the definition of *ring*:n#1 which is defined by *gold*:n#1.

Example 4: Is there a possible topical relation between *diamond* and *mine*?
Answer: there is a WW-path: *diamond*:n#1 ← *gem*:n#1 → *ring*:n#4 → *gold*:n#1 ← *mine*:n#1
Explanation: *gold*:n#1 is in the definition of *mine*:n#1, and the rest is as before.

Answering questions with summaries

There are questions whose answer cannot be directly mined from large text collections because the information they request is spread across full documents or across several documents. Instead of returning all those documents, the answer should summarize the textual information contained in the texts. To generate summaries as response to a question, the architecture of the typical QA system needs to be modified. Figure 3 illustrates the new QA architecture capable of answering question of the type *"What is the situation with the diamond market?"*.

Instead of capturing the semantics of the question by classifying against the question stem (e.g. *what*, *who* or *how*), the question processing module identifies the topic indicated by the question. Topic identification is performed by first ex-

[1] We denote a synset by one of its elements, followed by the part-of-speech of the synset and by the sense of that word. For parts-of-speech, we denote nouns as n, verbs as v, adjectives as a and adverbs as r, following the same notation as in WordNet.

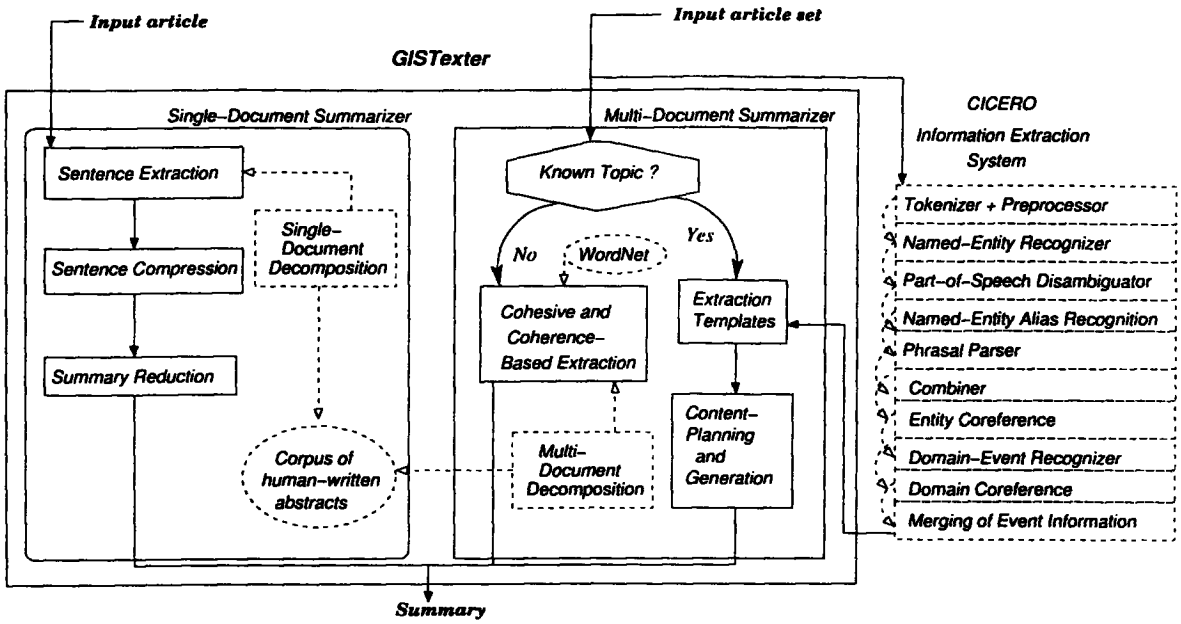

Figure 4: Architecture of GISTEXTER

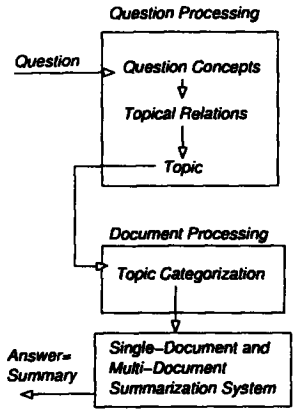

Figure 3: Architecture of QA systems that answer question with a summary.

tracting the context words from the question and then mining their associated topical relations from WordNet. Topical paths that display a high degree of redundancy are grouped as separate topics and are used to process the documents. Instead of performing paragraph retrieval as it is done in current fact-based QA system, document processing involves a topical categorization of the collection and returns either single documents that are characteristic of the topic or set of such documents. Finally, a summarization system generates the answer to the question in the form of a summary of predefined size or compression rate.

For the purpose of summarization, we have trained a summarization system on the corpus of documents and corresponding abstracts written by four different human assessors for the DUC-2001 evaluations. Our system, called GISTEXTER generates both single-document and multi-document summaries.

Two assumptions stay at the core of our techniques: (1) when generating the summary of a single document, we should extract the same information a human would consider when writing an abstract of the same document; (2) when generating a multi-document summary, we should capture the textual information shared across the document set. To this end, for the multi-document summaries we have applied the CICERO Information Extraction (IE) system to generate extraction templates for some of the topics that were already encoded in the multi-domain knowledge of CICERO[2].

For the topics that were not covered by CICERO we had a back-up solution of gisting information by combining cohesion and coherence indicators for sentence extraction. By definition, gisting is an activity in which the information taken into account is less than the full information content available, as reported in (Resnik 1997). In our backup solution, getting the gist of the summary information is performed by assuming that the more cohesive the gisted information is, the more essential it is to the multi-document summary. In addition coherence cue phrases were considered for information extraction. At the core of getting the gist of the summary stay topical relations extracted from WordNet.

The architecture of GISTEXTER is shown in Figure 4. Input to the system is either a single document or a set of documents. When a summary for a single document is sought, in the first stage key sentences are extracted, as in most of the current summarizers. A *sentence extraction* function is

[2]CICERO is an ARDA-sponsored on-going project that studies the effects of incorporating world knowledge into IE systems. CICERO is being developed at Language Computer Corporation.

learned, based on features of the *single-document decomposition* that analyzes the features of human-written abstracts for single documents. To further filter out un-necessary information, the extracted sentences are then *compressed*. In the last stage a *summary reduction* is performed, to trim the whole summary to the length of 100 words.

If multi-document summaries are sought, two situations arise:

Case 1: the topic of the document set has been encoded in the multi-domain knowledge of CICERO, therefore the IE system extracts templates filled with information relevant to the topic of the document set. To fill extraction templates CICERO performs a sequence of operations, implemented as cascaded finite-state automata. The texts are tokenized and the named entities are semantically disambiguated, then phrasal parses are generated and combined into noun and verb groups typical to a given topic of interest. Coreference between entities is resolved and the mentions of the events of interest are identified. Before merging information referring to the same event, coreference between entities related to the same event is resolved. The textual information that fills the templates is used for content planning and generation of the multi-document summary.

Case 2: the topic is new and unknown. In this case the human-written abstracts are decomposed to analyze the cohesive and coherence cues that determine the extraction of textual information for the multi-document summary. For the analysis of the cohesive information, we use the topical relations mined from WordNet as well as the semantic classes of the named entities provided by CICERO. The coherence information is indicated by cue phrases such as *because*, *therefore* or *but* but also by very dense topical relations between the words of the text fragment.

Single-Document Summarization

To generate 100-word summaries from single textual documents, we have implemented a sequence of three operations:

1. *Sentence Extraction* - identifying sentences in the original document that contain information essential for the summary;
2. *Sentence Compression* - reducing the extracted sentences by filtering out all non-important information while preserving the grammatical quality;
3. *Summary Reduction* - trimming the resulting summary to the required 100 word/document limit.

The sentence extraction is based on the topical relations, whereas sentence compression combines the topical relations with syntactic dependencies. The summary reduction simply retains only the first 100 words from the summary.

Multi-Document Summarization

The generation of multi-document summaries is a two-tired process. When the topic is one of the 20 topics encoded in CICERO, we generate the summary as a sequence of three operations:

1. *Template Extraction* - filling relevant information in the slots of predefined templates corresponding to each topic;
2. *Application of Summary Operators* - after common features of multiple templates are identified and the distinct features are marked up, summary operators are applied to link information from different templates. Often this results into the synthesis of new information. For example, a generalization may be formed from two independent facts. Alternatively, since almost all templates have a TIME and a LOCATION slot, contrasts can be highlighted by showing how events changes along time or across different locations. The operators we encoded in GISTEXTER are listed in Table 3 and were defined and first introduced in (Radev and McKeown 1998).
3. *Ordering of Templates and Linguistic Generation* - whenever an operator is applied, a new template is generated by merging information from the arguments of the operator. All such resulting templates are scored highly then the extracted templates. Moreover, depending on the slots the operators merged, the scores have different values. At the end, information from templates with higher importance appear with priority in the resulting summary. We use the topical relations to realize only those sentence fragments that are identified by extraction patterns.

Name	Definition
Change of Perspective	The same event described differently
Contradiction	Conflicting information about the same event
Addition	Additional information about an event is brought forward
Refinement	More general information about an event is refined later
Superset/Generalization	Combine two incomplete templates
Trend	Several templates have common fillers for the same slot

Table 3: Summary Operators.

When the topic is not encoded in CICERO, we need to further implement two additional steps:

1. *Automatic Generation of the Extraction Template* - in which topical relations are used to find redundant information, and thus use it as slots of the template;
2. *Automatic Acquisition of Extraction Patterns* - in which linguistic rules capable of identifying information relevant to the topic in text and to extract it by filling the slots of the template. Usually, such textual information represents only about 10% of a document.

For generating extraction templates we collect all the sentences that contain at least two concepts that are related to the topic. We then categorize semantically all these concepts and select the most common classes observed in the corpus. To these classes we add Named Entity classes (e.g. PERSON, LOCATION, DATE and ORGANIZATION). We consider the each of these two types of classes as the slots of the template.

When the extraction template is ready, extraction patterns are learned by applying the bootstrapping techniques de-

scribed in (Riloff and Jones 1999). If there is any document from which we cannot extract at least a template, we eliminate the slot corresponding to the least popular semantic class and continue until we have at least on template per document.

The extraction techniques have the role of pointing where the information that should be included in the summary exists. But we also need to paste this information in a coherent and cohesive summary. This form of generation was first introduced in (Jing and McKeown 2000). Instead of using knowledge represented in the form of templates alone, cut-and-paste operators provide a way of generating summaries by transforming sentences selected from the original documents because they were matched by extraction patterns. The cut-and-paste operators are listed in Table 4.

Name	Definition
Sentence reduction	Remove extraneous phrases from sentence
Sentence combination	Merge text from several sentences
Paraphrasing	Replace phrases with their paraphrases
Generalization/Specification	Replace phrases with more general/specific phrases
Reordering	Change the order of sentences

Table 4: Cut-and-Paste Operators.

Conclusions

In this paper we described a new paradigm for answering natural language questions. Whenever a question relates to a topic that is discusses extensively in a document or set of documents, the answer produced is a summary of those texts. The generation of the summary relies on mining topical relations from the WordNet knowledge base. In this way, answers are mined both from the texts and the lexical knowledge base encode in WordNet.

Acknowledgments: This research was partly funded by ARDA's AQUAINT program.

References

Christiane Fellbaum (Editor). WordNet - An Electronic Lexical Database. MIT Press, 1998.

S.J. Green. Building Hypertext Links by Computing Semantic Similarity. *IEEE Transactions on Knowledge and Data Engineering*, pp 713-730,Sept/Oct 1999.

S.M. Harabagiu and D.I. Moldovan. Knowledge Processing on an Extended WordNet. WordNet-An Electronic Lexical Database. The *MIT Press*, C. Fellbaum editor, pp 379-406, 1998.

S. Harabagiu, G.A. Miller and D. Moldovan. WordNet 2 - A Morphologically and Semantically Enhanced Resource. In *Proceedings of SIGLEX-99*, June 1999, Univ. of Maryland, pages 1-8.

S. Harabagiu and S. Maiorano. Finding answers in large collections of texts: Paragraph indexing + abductive inference. In *AAAI Fall Symposium on Question Answering Systems*, pages 63–71, November 1999.

S. Harabagiu, M. Paşca and S. Maiorano. Experiments with Open-Domain Textual Question Answering. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000)*, August 2000, Saarbruken Germany, pages 292-298.

S. Harabagiu, D. Moldovan, M. Paşca, R. Mihalcea, M. Surdeanu, R. Bunescu, R. Gîrju, V. Rus and P. Morărescu. The role of lexico-semantic feedback in open-domain textual question-answering. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001)*, Toulouse, France, 2001, pages 274–281.

G. Hirst and D. St-Onge. Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms. WordNet-An Electronic Lexical Database. The *MIT Press*, C. Fellbaum editor, pp 305-332, 1998.

E. Hovy, L. Gerber, U. Hermjakob, C-Y. Lin, D. Ravichandran. Toward semantic-based answer pinpointing. In *Proceedings of the Human Language Technology Conference (HLT-2001)*, San Diego, California, 2001.

H. Jing and K. McKeown. Cut and Paste Text Summarization. In *Proceedings of the First Meeting of the North American Chapter of the Association of Computational Linguistics (NAACL-2000)*, pages 178–185, 2000.

W. Lehnert. *The Process of Question Answering: a computer simulation of cognition*. Lawrence Erlbaum, 1978.

G.A. Miller. WordNet: a lexical database. *Communications of the ACM*, 38(11):39–41, 1995.

J. Morris and G. Hirst. Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text. In *Computational Linguistics*, Vol. 17 , no 1, pp. 21-48, 1991.

D.R. Radev and K.R. McKeown. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):469–500, 1998.

P. Resnik. Evaluating Multilingual Gisting of Web Pages. *AAAI Spring Symposium on Natural Language Processing for the World Wide Web*, 1997.

E. Riloff and R. Jones. Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping. Proceedings of the *Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, pages 474-479, 1999.