# Answering Complex Questions with Random Walk Models

Sanda Harabagiu, Finley Lacatusu and Andrew Hickl
Language Computer Corporation
1701 N Collins Blvd. Suite 2000
Richardson, TX 75080
{sanda, finley, andy}@languagecomputer.com

## ABSTRACT

We present a novel framework for answering complex questions that relies on question decomposition. Complex questions are decomposed by a procedure that operates on a Markov chain, by following a random walk on a bipartite graph of relations established between concepts related to the topic of a complex question and subquestions derived from topic-relevant passages that manifest these relations. Decomposed questions discovered during this random walk are then submitted to a state-of-the-art Question Answering (Q/A) system in order to retrieve a set of passages that can later be merged into a comprehensive answer by a Multi-Document Summarization (MDS) system. In our evaluations, we show that access to the decompositions generated using this method can significantly enhance the relevance and comprehensiveness of summary-length answers to complex questions.

## Categories and Subject Descriptors

H.3.m [**INFORMATION STORAGE AND RETRIEVAL**]: Miscellaneous; I.2.7 [**ARTIFICIAL INTELLIGENCE**]: Natural Language Processing

## General Terms

Algorithms, Performance, Experimentation

## Keywords

Question Answering, Summarization

## 1. INTRODUCTION

Complex questions cannot be answered using the same techniques that apply to "factoid" questions. Complex questions refer to relations between entities or events; they refer to complex processes and model scenarios that involve deep knowledge of the topic under investigation. For example, a question like $Q_0$: "*What are the key activities in the research and development phase of creating new drugs?*" looks for information on two distinct phases of creating drugs. Typically, relevant information for these kinds of questions can be found in multiple documents and needs to be fused together into a final answer. In the Document Understanding Conferences (DUC), the answer to complex questions like $Q_0$ is considered to be a multi-sentence multiple document summary (MDS) that meets the information need of the question. We introduce a new paradigm for processing complex questions that relies on a combination of (a) question decompositions (of the complex question); (b) factoid question answering (Q/A) techniques (to process decomposed questions); and (c) multi-document summarization techniques (to fuse together the answers provided for each decomposed question). Central to this process is a question decomposition model that enables the selection of the textual information aggregated in the final answer.

We present a novel question decomposition procedure that operates on a Markov chain model inspired from the Markov chains used for expanding language models introduced in [9]. We propose that question decomposition depends on the successive recognition (and exploitation) of the relations that exist between words and concepts extracted from topic-relevant sentences. (For the purposes of this paper, we will define a relation as any semantic property that can exists between two or more entities or events in texts.) For example, if a topic-relation $r_1$ between "develop" and "drugs" is recognized in question $Q_0$, we assume that this sentence (and all other sentences containing this particular relation) will contain relevant information that can be used to decompose of $Q_0$. Furthermore, we expect that sentences containing topic-relevant relations will also contain other relevant relations that should be leveraged in question decomposition. For example, if $r_1$ is identified in the sentence $s_1$ "*The challenge for Glaxo was to develop a drug that was pleasant to swallow.*", we expect that a new relation $r_2$ between the concept COMPANY ("Glaxo") and "develop" should be extracted and used to identify still other sentences that could potentially provide relevant information. As new relations are discovered, we expect that sentences containing the most relevant relations (or combinations of relations) can be used to generate questions that can represent possible decompositions of the original complex question. For example, given $r_1$ and $r_2$ in $s_1$, a question like "*What companies develop new drugs?*" can be created which could be used to obtain a set of answers which could represent a partial response to $Q_0$. Relevant answers to each newly-decomposed question can be used to discover more relevant relations, that in turn, prompt still more question decompositions. This process ends when either no new relations are discovered, or the random walk is stabilizing within a threshold.

We evaluate question decompositions in three ways. First, we compare them against decompositions produced by humans. Second, we conduct several evaluations of the quality of the MDS answers they enable. Third, we use every sentence from the MDS

*SIGIR'06,* August 6–11, 2006, Seattle, Washington, USA.
Copyright 2006 ACM 1-59593-369-7/06/0008 ...$5.00.

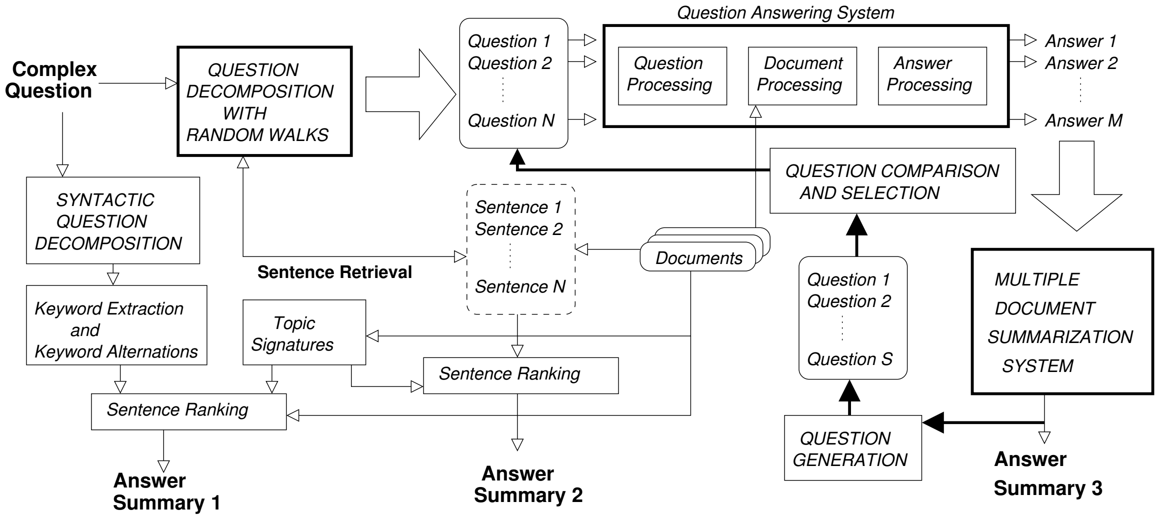

**Figure 1: The Architecture of our Framework for Processing Complex Questions.**

answer and generate questions with the same procedure employed when creating question decompositions from relevant sentences. The questions that have answers in the summaries are evaluated against questions generated by human linguists. They are also used for measuring the similarity to the decomposed questions. Our studies indicate that these comparisons correlate with the relevance of the answers. We claim that this is an important finding since current MDS evaluation methods typically rely on (a) human produced answers, or (b) human judgments. The automatic scoring of the MDS answers based on comparisons of decomposed questions allows a framework in which researchers can test multiple Q/A techniques or multiple MDS techniques that best operate for finding answers.

The remainder of the paper is organized as follows. Section 2 presents the framework we have designed for processing complex questions. Section 3 details the question decomposition procedure. Section 4 describes the random walk models employed for decomposing questions. Section 5 details the evaluation results while Section 6 summarizes the conclusions.

## 2. PROCESSING COMPLEX QUESTIONS

In this section, we outline three methods for producing answers to complex questions from based on the output of a question decomposition system. By decomposing a complex question into a set of simpler subquestions that each represent a different dimension of the complex question's information need, we expect to be able to identify answers that are both informative and responsive.

In this paper, we introduce a new technique for question decomposition that uses a random walk in order to generate possible decompositions of a complex question. Figure 1 illustrates the system described in this paper.

Figure 1 includes two types of question decomposition modules: a syntactic question decomposition module and a random walk-based question decomposition module. With syntactic question decomposition, overtly-mentioned subquestions are extracted from a complex question by separating conjoined phrases and recognizing embedded questions. While syntactic decomposition is an important part of any question decomposition algorithm, we will not be discussing techniques for this type of decomposition in this paper. [1]

After complex questions are decomposed syntactically, as illustrated in Figure 1, keywords are extracted from each sub-question and are expanded with keyword alternations. The keywords are expanded by (1) identifying the semantic class to which they belong, and (2) using other terms from the lexicons associated with such semantic classes. To identify the semantic class, the keyword is matched against the lexicon of the class. The keyword alternations are selected from the first 20 scored words from the lexicon. The semantic classes are acquired off-line with a co-training method reported in [18].

**research:** trial, effort, step, study, work, activity, area, business, cause, field, function, issue, program, project, sector, service, site, education, information, science

**drug:** amphetamine, cocaine, ecstasy, epo, heroin, lsd, marijuana, medication, morphine, opium, measure, prozac, ritalin, steroid, treatment, viagra, alcohol, cost, disease, issue

**Figure 2: Keyword Alternations.**

Figure 2 illustrates the keyword alternations resulting for the keywords "research" and "drug" that were extracted from the subquestion "*What are the key activities in the research phase of creating new drugs?*".

Additionally, we use two different models of topic signatures to identify (a) the most representative relations for the topic referred by the complex question and evidence by the document collection. The first topic signature ($TS_1$) we have implemented was reported in [10]. $TS_1$ is defined by a set of terms $t_i$, where each term is highly correlated with the *topic* with an associated weight $w_i$: $TS_1 = \{topic\langle(t_1, w_1), (t_2, w_2), \ldots, (t_n, w_n)\rangle\}$. The selection of the terms for $TS_1$ as well as the assignment of the association weights is determined by the use of the likelihood ratio. The second topic signature ($TS_2$) was introduced in [3]. It takes into account the fact that topics are not characterized only by terms, there are also relations between topic concepts that need to be identified. If only nouns and verbs from $TS_1$ are selected as topic concepts, the topic signature $TS_2$ is defined as $TS_2 = \{topic, \langle(r_1, w_1), (r_2, w_2), \ldots (r_n, w_n)\rangle\}$, where $r_i$ is a binary relation between two topic concepts. The procedure of generating $TS_2$ was detailed in [3], and it identifies two forms of relations: (a) syntax-based relations, and (b) salience-based context relations. The arguments of these relations may be (1) nouns or nominalizations; (2) named entity types that a Named Entity Recognizer (NER) identifies; and (3) verbs.

[1] For more information on syntactic decomposition, see [8].

When topic signatures are available, each sentence from the document collection receives a score based on (a) the presence of a term from $TS_1$; (b) the presence of a relation from $TS_2$; and (c) the presence of any of the keywords extracted from the sub-question or their alternations. The sentence scores determine a ranking of the sentences from the collection for each sub-question. Finally, the answer is produced by selecting for each decomposition only the corresponding highest ranked sentences. Redundancy is eliminated by checking that each new added sentence does not contain any predicate-argument relation that was already present in a previously selected sentence. Predicate-argument relations are discovered by processing sentences with a semantic parser trained on PropBank [16]. Additionally, each predicate and argument is mapped into every WordNet synonym to enable paraphrase identification. In this way Answer Summary 1 from Figure 1 is produced.

In addition to the method described above, complex questions can also be decomposed by another method that is described in Sections 3 and 4. Due to this, in our framework we can produce two additional answers as summaries. The second form of question decomposition discovers relations relevant to the complex question and sentences in which they are present. For each such sentence, one or multiple questions are generated, representing additional question decompositions. When these decompositions are ignored and only the sentences are considered, the topic signatures can be used to score them and to produce a second answer as summary (Answer Summary 2 illustrated in Figure 1).

When complex questions are decomposed using random walks, subquestions are submitted a state-of-the-art question-answering (Q/A) system (described in [5]), which returns sets of ranked relevant answers for each such decomposition. All these answers are considered separate documents, which are used to produce a multi-document summary as the third answer (MDS) (Answer Summary 3 illustrated in Figure 1). The MDS system that was used has been reported in [7]. Furthermore, for each sentence in the third answer, we generate one or several questions with the same technique that is used for decomposing questions with random walks. Since questions produced from the complex question, and questions produced from the answer are available, we argue that the answer is relevant if the two sets of questions are very similar. Question comparison is produced by a battery of four question similarity measures, previously reported in [4]. In Section 5 of this paper we detail the similarity measures we used in the experiments. The selection of only the most similar questions improves the quality of the answer. Instead of submitting all questions generated by the random walks, only the most similar questions are processed again by the Q/A system, thus closing a feedback loop. Using a hill-climbing technique, if the aggregate similarity of the new set of questions derived from the new answer is improved significantly, the feedback loop starts again. The aggregate similarity is also described in Section 5 of this paper.

The feedback loop ends either when new improvements are not obtained, or when the number of loops is larger than a threshold, in our case $L_T = 7$. With this framework , we were able to study the effects of different forms of question decompositions on the quality of the answers.

## 3. DECOMPOSING COMPLEX QUESTIONS

In order to process complex questions like $Q_0$: "*What are the key activities in the research and development phase of creating new drugs?*", current Q/A systems need to decompose the question in a series of simpler questions, that can be tackled by the factoid-based techniques that have emerged from the TREC Q/A evaluations. Table 1 illustrates some of the questions that represent decompositions of $Q_0$ and can be generated automatically by the technique we present in this section.

| | |
|---|---|
| $Q_0^1$: | *What companies develop new drugs?* |
| $Q_0^2$: | *What diseases are new drugs being developed for?* |
| $Q_0^3$: | *How long does it take to develop a new drug?* |

**Table 1: Examples of Question Decompositions.**

The main feature of the decomposed question is related to the ability to easily detect their *expected answer type* (EAT), which represents the semantic class to which their answers should belong. For example, the EAT of $Q_0^1$ is ORGANIZATION, the EAT of $Q_0^2$ is DISEASE, whereas the EAT of $Q_0^3$ is DURATION. Our main assumption is that the question decomposition model should be based on several types of relations between words or concepts used in (a) the complex question, (b) in sentences that contain relevant information for the complex question, or (c) in other question decompositions that have been produced before for the same complex question.

In order to produce question decompositions, we follow four steps. In the first step we process the complex question for deriving the relations that are meaningful. In the second step we generate questions based on the relations selected. In the third step we enhance the meaningful set of relations with relations discovered when generating a question decomposition and then we select a new relation based on the latest decomposition. In the fourth step, we loop back to step 2 unless the probability to continue is not above a certain threshold. The detailed operations in each step are:

**STEP 1:** The complex question is lexically, syntactically and semantically analyzed with the goal of identifying the relationships between words that may lead to the generations of simpler questions. The three forms of knowledge are marked up in each of the phases of the analysis:

**1.a.** (*lexical*) The determination of the part-of-speech of each word, generated by the Brill tagger [1].

**1.b.** (*syntactic*) A full parse of the question is generated by the probabilistic parser reported in [2]. The result of the parse renders information about the syntactic constituents of the question and about their relations. For example, for the complex question $Q_0$, we derive the following constituents: $VP_1$: {*are*}; $VP_2$: {*containing*}; $NP_1$: {*the key activities*}; $NP_2$: {*the research*}; $NP_3$: {*development phase*}; $NP_4$: {$NP_2$ *and* $NP_3$}; $NP_5$: {*new drugs*}; $PP_1$: {$NP_4$ *of* $NP_5$}; $PP_2$: {$NP_1$ *in* $PP_1$} [2].

**1.c.** (*lexical*) For each base NP (e.g. $NP_1$, $NP_2$, $NP_3$, $NP_5$) we determine whether the head is a nominalization of some verb, by accessing the WordNet database [12]. For example, the noun "*development*" is a morphological derivation of the verb "*develop*". The NPs having heads which are nominalizations are not considered in Step 1.d.

**1.d.** (*lexical/semantic*) The generality of the heads of each NP is assessed in one of the two categories: *abstract*, or *concrete*. The assessment is based on a large answer type taxonomy that was developed for the TREC evaluations of Q/A systems. The taxonomy, which was described in [17] comprises 440 synsets from WordNet (and their hyponyms) and 150 semantic classes of names that are recognized by the Named Entity Recognizers we have available. If any of the heads of an NP is found in the answer taxonomy, it is assigned the attribute *concrete*, otherwise it is labeled *abstract*. For the question $Q_0$, only the head of $NP_5$ is categorized as *concrete*. $NP_1$ is labeled *abstract*. The question processing techniques ap-

[2] NP stands for noun phrase, VP for verb phrase, and PP for prepositional phrase

plied for factoid Q/A identify $NP_1$ as being the constituent that indicates the EAT for the question. Since no EAT can be established for $Q_0$, it is considered a complex question.

**1.e.** (*syntactic*) Relations between concrete NPs and other constituents are sought. The syntactic relationship from the constituent $PP_1$ indicates a prepositional attachment relation between $NP_5$ and $NP_4$, which is a coordination between $NP_2$ and $NP_3$. The syntactic decomposition of the coordination entails two relations between the verbs related to $NP_2$ and $NP_3$ and $NP_5$. The output of Step 1 for $Q_0$ is: RELATIONS: $\{R_1$ = [develop – new drugs]; $R_2$ = [research – new drugs]$\}$

**STEP 2:** For a relation discovered at Step 1 we generate questions that involve that relation. In order to generate questions automatically, we employ a method that was first reported in [4]. In order to generate the question, we first find a sentence that constitutes an answer for that question. This is done by the following sub-steps:

**2.a.** Query Formulation. In order to find sentences in which elements from the RELATIONS list are discovered, we formulated two kinds of queries: (a) queries involving the lexical arguments of the relation, e.g. ["develop" AND "drug"] as well as (b) queries that involved semantic extensions. Four forms of extensions were considered: (1) extensions based on the semantic class of names that represent the nominal category (e.g. names of drugs), (2) extensions based on verbs which are semantically related to the verb in the WordNet database (e.g. develop(v) –sem.relation→ create(v); develop(v) –sem.relation→ produce(v)); (3) extensions that allow the nominal to be anaphoric, therefore replaced by a pronoun, e.g. [develop – it]; and (4) extensions that allow the nominalizations, as well as the verbal conjuncts, to be considered.

| | |
|---|---|
| $r_j$ shares a predicate with $r_i$ | $r_j$ shares an argument with $r_i$ |
| $r_j$ specializes the predicate of $r_i$ | $r_j$ specializes the argument of $r_i$ |
| the predicates of $r_j$ and $r_i$ can be composed | |

**Table 2: Properties Between Relations $r_i$ and $r_j$.**

**2.b.** Sentence Retrieval. We built an index based on the processing of relations in the text collection[3]. A sentence is added to the inverted list of a relation $r_i$ when it may be composed with another relation $r_j$ in the same sentence and (a) relations $r_i$ and $r_j$ meet one of the conditions listed in Table 2, or the predicates of relations $r_i$ and $r_j$ may be composed with the predicate composition procedure described as a special case in 2.c; and (b) the argument of the relation $r_j$ can be mapped in one of the EAT categories of the Q/A system. Examples of such sentences are illustrated in Table 3. Sentence $S_1$ is retrieved because it contains relation $r_i$ = [develop – drugs] and also a relation $r_j$ = [develop – Glaxo] that shares the same predicate ("develop") and "Glaxo" is mapped into the EAT = COMPANY. Similarly, sentences $S_2$, $S_3$, and $S_4$ are retrieved because they contain three different expansions of $r_i$ and new relations that are compatible with it.

**2.c.** Question Generation. Every sentence retrieved at 2.b. contains additional relations, besides those that were expressed by the query. Among those relations, some share arguments with the queried relations, some do not. The first group of relations may serve to point to EATs that the decomposed questions should refer to. For example, in sentence $S_1$, the new relation [Glaxo – develop] can be generalized into [ORGANIZATION – develop] in which ORGANIZATION can be selected as the EAT of the question that shall be generated. Our named entity recognizer (NER) is able to distinguish between different types of organizations, tagging "Glaxo" from sentence $S_1$ as COMPANY, and "Medical School" from $S_2$ as UNIVERSITY. When the EAT is established, the question stem that is associated with it is known (e.g. "what companies") and it substitutes the name from the sentence, to generate the question $Q_0^1$ from Table 1, in which relation $R_1$ is fully specified with all the argument adjuncts it had in the complex question $Q_0$. Sentence $S_1$ generates the question $Q_0^1$, whereas sentence $S_4$ generates the question "*What universities develop drugs?*". Sentence $S_5$ illustrated in Table 3 enables the generation of $Q_0^2$, whereas sentence $S_6$ is used for generating $Q_0^3$. Starting from relation $R_1$ in RELATIONS, three new relations are discovered: $R_1^1$ = [[$R_1$ = develop – drug] – COMPANY], $R_1^2$ = [[$R_1$ = develop – drug] – DISEASE], and $R_1^3$ = [[$R_1$ = develop – drug] – DURATION]. Each of these new relations enable the generation of the decomposed questions listed in Table 1.

| | |
|---|---|
| $S_1$: | *The challenge for Glaxo was to develop a drug that was pleasant to swallow.* |
| $S_2$: | *The remaining 60 per cent of royalties will be paid to [...] Charing Cross and Westminster Medical School which developed it.* |
| $S_3$: | *Few companies admit setting out to create me-too drugs.* |
| $S_4$: | *Cancer Research funded research and development of the drug which was originally discovered by Aston University in Birmingham.* |
| $S_5$: | *At Bristol-Myers, which he left in 1980 to join SmithKline, Crooke helped develop an array of chemotherapy drugs for cancer patients that put Bristol at the forefront of cancer treatment.* |
| $S_6$: | *Since a typical drug takes 10 years and Y10bn to develop, only those companies large enough to absorb the costs will be able to survive.* |

**Table 3: Sentences retrieved for relation $R_1$: [develop – drug].**

◇ Special Cases. The properties between relations $r_i$ and $r_j$ that are used in the index cover three more cases that need to be addressed by question generation. They are:

**2.c.** Argument Specialization. In order to inquire about the attributes of arguments, three forms of questions are generated: (i) questions that inquire about instances of entities that are referred by the argument of a relation in which the semantic class of the argument is the EAT of the question, and the question becomes a list question; (ii) questions that specialize the argument of the relation by using a modifier which becomes the EAT of the question; and (iii) questions that inquire about the characteristics of the arguments by using the question stem "what types". An example of the first form of questions is $Q_0^6$: "*What new drugs have been developed?*", generated from the sentence: "*Zinnat is a new drug which was developed because other drugs in its class needed to be injected and were therefore of little use outside the hospital environment.*". An example of the second form of questions is $Q_0^7$: "*How many medicines are launched per year?*" , in which the EAT is NUMBER, it modifies the argument "medicines" in sentence "*The number of medicines launched during the early 1980s averaged about 60 per year.*".

| |
|---|
| *How are new drugs researched?* |
| *How are drugs manufactured?* |
| *What types of activities are included in the development of new drugs?* |

**Table 4: Questions Based on Predicate Specialization.**

**2.d.** Predicate Specialization. There are three ways of specializing the predicates from the relations: (i) by selecting the EAT of the question as a MANNER, and associating the question stem "how"; (ii) by using adjuncts of the predicates in the question to produce either a specialized MANNER EAT or a YES/NO question; and (iii) by considering that the predicates represent complex events that have structure, and thus this structure can be inquired by using special constructs of the form "what steps are included in", or "what types of activities are included in". The first form of predicate specialization is the most productive one, and it can be generated based on the recognition of MANNER relations that was reported in [6].

[3] We process the text collection and discover all syntactic and salient relations when we build the topic signature $TS_2$ described in Section 2.

Examples of questions that were generated for predicate specialization are listed in Table 4.

**2.e.** Predicate Composition. Some questions need to capture relations between predicates. Such relations may be determined by the discovery of (a) causal relations, as it was reported in [4]; (b) temporal relations; or (c) because the predicates share an argument. Table 5 illustrates such questions, their type of relations, and the sentences that enabled them. In our implementation, we have used a set of cue phrases and causal verbs to detect causal relations between predicates. For the temporal relations, we relied on the temporal signals annotated in TimeBank (e.g. "before", "after", "during").

| |
|---|
| CAUSAL:<br>*How do trade restrictions affect new drug development?* |
| TEMPORAL:<br>*How many times must a drug be tested before it can be sold?* |
| ARGUMENT SHARING:<br>*How do companies decide which new drugs to research?* |

**Table 5: Questions Based on Relations Between Predicates.**

**STEP 3:** The selection of a new relation is performed after newly discovered relations are added to the RELATIONS list.

**3.a.** Relations that specialize arguments or predicates are not added to RELATIONS, but all the other three types of relations $r_j$ are appended. For example, the relations $R_1^1$ = [[$R_1$ = develop – drug] – COMPANY], $R_1^2$ = [[$R_1$ = develop – drug] – DISEASE], and $R_1^3$ = [[$R_1$ = develop – drug] – DURATION] are added.

**3.b.** A new relation is selected to maximize the probability estimation that it will lead to another question decomposition of the complex question. The probability estimations are detailed in Section 4.

**STEP 4:** The decision to continue or stop the process of generating question decompositions depends on our formalization of the process. We have formalized the process of generating question decompositions which lead to the discovery of new meaningful relations as a random walk on a bipartite graph of questions and relations. For a given relation, a sentence that contains the relation is selected. That sentence is considered to be the answer to a question decomposition, which is generated by identifying a new relation, which in turn, when selected will lead to a new question decomposition. Thus the random walk continues with a probability $\alpha$, generating a new decomposition and selecting a new relation, or it stops with a probability $(1-\alpha)$. Section 4 describes the formalisms that allow us to estimate the probability that the random walk ends after $k$ steps, corresponding to $k$ loops of the Steps 2 and 3 of this procedure.

# 4. MARKOV CHAINS FOR QUESTION DECOMPOSITION

In this section, we describe how we employ two different types of random walks to decompose complex questions for question-answering and/or multi-document summarization applications. We begin by describing how a random walk can be used to populate a network with potential decompositions of a complex question. Later, in Section 4.2, we show how another random walk can then be used to select a set of generated decomposed questions that best represents the information need of the complex question.

The question decomposition procedure detailed in Section 3 can be cast as a Markov chain (MC). A MC over a set of states $S$ is specified by an initial distribution $p_0(S)$ over $S$, and a set of state transition probabilities $p(S_t|S_{t-1})$. In the case of question decomposition, the initial state is represented by one relation $r_0$ selected from the list RELATIONS ($time = 0$), which is the set of relations generated when processing the complex question. The probability of the initial state is set as $\frac{1}{n}$, where $n$=|RELATIONS($time = 0$)|. After selecting a relation $r_i$ at step $i$, the index is consulted to find sentences where $r_i$ and other relations $r_j$ having the properties listed in Table 2 are present. If the argument of relation $r_j$ can be categorized in the EAT hierarchy as an expected answer type $e_j$, then it can enable the generation of a question decomposition $QD_{i+1}$ with EAT = $e_j$. The probability that a question decomposition $QD_{i+1}$, with EAT = $e_j$, is generated from a relation $r_i$ is given by $p(QD_{i+1}|r_i) = p(e_j|r_i)$. The new relation $r_j$ is placed in the RELATIONS list. If the index of $r_i$ had only one sentence and only one relation $r_j \neq r_i$ could be found in that sentence, then $r_i$ is removed from the list RELATIONS.

A new relation $r_{i+1}$ is selected from RELATIONS based on the probability $p(r_{i+1}|QD_{i+1})$. Since question $QD_{i+1}$ was generated based on the EAT discovered with the help of relation $r_j$ which led to the EAT $e_j$, we can evaluate $p(r_{i+1}|QD_{i+1}) = p(r_{i+1}|e_j)$. In this way we have defined the transition probabilities of the Markov Chain (MC) illustrated in Figure 3. The MC alternates from selecting relations from RELATIONS and generating a new question decomposition. In this way, the decomposition process is "surfing" the set of relations meaningful for the complex question, and also the decomposed questions that are generated based on these relations. After each step there is some chance that the question decomposition process will stop. The process continues the random walk with probability $\alpha$, generating a new set of question decompositions. With probability $(1 - \alpha)$, the walk stops after step $k$ (after producing the question decomposition $QD_{k+1}$).

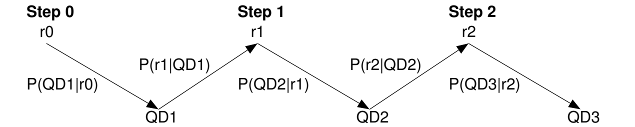

**Figure 3: The Markov chain alternates between relations and question decompositions.**

Since our goal is to estimate the probability that the MC stops after $k$ steps, we produce a matrix formulation of the problem which is similar to the formulation reported in [9]. This formulation is described in Section 4.1. We also want to test our hypothesis that the decomposed questions are relevant for the complex question. Since these question decompositions have been generated by relations that we have discovered in the text to be associated with relations originating in the complex question, we want to test if our assumption that they are valid decompositions can be quantified by a measure of relatedness to the complex question. For this purpose, in Section 4.2 we define a mixture model which generates a different random walk that evaluates the relevance of the decomposed questions.

## 4.1 Matrix Formulation

The operation of the random walk can be cleanly described by using a matrix notation. Let $N$ be the size of the index. The number $N$ corresponds to the relations that we have discovered in the text, having the properties that for every relation $r_i$ there is also a relation $r_j$ sharing with $r_i$ the properties from Table 2, and $r_j$ has the argument mapped in one of the semantic categories of the EAT classes. Let $M$ be the number of EAT classes. We consider $A$ to be a $N \times M$ stochastic matrix with entries $a_{ij} = p(r_i|e_j)$ representing the probability that the relation $r_i$ will be composed in a sentence with a relation $r_j$ that has an argument of semantic type $e_j$, which will become the EAT of the question decomposition that is generated. Similarly, a stochastic matrix $B$ of dimensions $M \times N$ can be defined, in which the elements $b_{ij} = p(e_i|r_j)$ represent the probability that a sentence that contains the relation $r_j$ can be the answer to a question with the EAT = $e_i$. Then, the $N \times N$ stochastic matrix

$C$ is defined as $C = A \times B$. The probability that the MC stops after $k$ steps is given by $(1-\alpha)\alpha^k C^k_{r,e}$, if the last relation it discovered is $r$ and the last question decompositions it has generated had the EAT = $e$.

To estimate $p(e_i|r_j)$ we consider

$$p(e_i|r_j) = \frac{p(r_j|e_i)p(e_i)}{\sum_k p(r_k|e_i)p(e_i)}$$

where $p(e_i)$ is the prior distribution of the semantic type $e_i$ in the corpus, and $p(r_j|e_i)$ is given by the maximum likelihood estimate of the relation distribution in the text collection. Let $J_1$ be the number of instances of the relation $r_j$ composed with a relation $r_i$ in the same sentence such that the argument of $r_i$ has the semantic type $e_i$. Then, $p_{mle}(r_j|e_i) = \frac{J_1}{\#(instances\ of\ r_j)}$

## 4.2 Random Walks with Mixture Models

Recently, [15] introduced a random walk model for finding answers to complex questions. This model is based on the idea that answers can be found by scoring each sentence against a complex question and selecting only the first top-ranked sentences. The sentence rank is produced by a mixture model that combines an approximation of a sentence's relevance to a question with similarity measures that can be used to select answer sentences that are not similar to one another. Using the same idea, we devised a similar mixture model for measuring the relevance of a question decomposition $qd_i$ to the complex question $cq$. The relevance measure is defined as:

$$\begin{aligned} relevance(qd_i, cq) &= d\frac{sim_a(qd_i, cq)}{\sum_{qd_j} sim_a(qd_j|cq)} + \\ &+ (1-d)\sum_{qd_j}\frac{sim_b(qd_i, qd_j)}{\sum_{qd_k} sim_b(qd_k, qd_j)} \end{aligned}$$

The similarities $sim_a$ and $sim_b$ are selected from the four similarity measures defined in Section 5. If $k$ is the number of question decompositions that we consider, a stochastic matrix $A$ of dimensions $k \times k$ is considered such that $a_{ij} = \alpha * relevance(qd_i, cq)$. In order for matrix $a$ to be stochastic, $\sum_i a_{ij} = 1$, thus $\alpha = 1/\sum_{i=1}^{k} relevance(qd_i, cq)$. Similarly, a stochastic matrix $B$ of dimension $k \times k$ is defined such that $b_{ij} = \beta_j * sim_b(qd_i, cq)$, where $\beta_j = 1/\sum_{i=1}^{k} sim_b(qd_i, qd_j)$. Next, the relevance vector $R$ for all question decompositions is defined by $R = [dA + (1-d)B]_i \times R$. The square matrix $E = [dA + (1-d)B]$ defines a MC where each element $e_{ij}$ from $E$ specifies the transition probability from state $i$ to state $j$ in the corresponding Markov Chain. The relevance vector $R$ is the stationary distribution of the Markov chain. With probability $d$, a transition is made from the current question decomposition $qd_i$ to new question decompositions that are similar to the complex question $cq$. With a probability $(1-d)$, a transition is made to question decompositions that are similar to the last question generated, $qd_i$. We have used several values for $d$ in our experiments.

# 5. EVALUATION RESULTS

Our experiments have targeted (1) the evaluation of the decomposed questions; (2) the evaluation of the three forms of answers produced by the framework illustrated in Figure 1; and (3) the evaluation of the impact of the decomposed questions on the quality of answer summaries.

**Evaluation of Decomposed Questions**

The evaluation of the decomposed questions was performed in two ways. First, the decomposed questions were evaluated against decompositions created by humans. Second, question decompositions were evaluated against questions generated from the answer summaries. The second evaluation was also compared against an evaluation involving only human-generated questions, both from the complex question and from the answer summaries. The evaluation was performed against 8 complex questions that were asked as part of the DUC 2005 question-directed summarization task. The questions correspond to the topics listed in Table 6.

Four human annotators performed manual question decomposition based solely on the complex questions themselves. Annotators were asked to decompose each complex question into the set of subquestions they felt needed to be answered in order to assemble a satisfactory answer to the question. (For ease of reference, we will refer to this set of question decompositions as $QD_{human}$.)

In order to evaluate the quality of the automatic question decompositions produced by our system, we generated three different types of question decompositions for a total of 8 complex questions that were asked as part of the 2005 DUC question-directed summarization task. First, we had 4 human annotators perform manual question decomposition based solely on the complex questions themselves. Annotators were asked to decompose each complex question into the set of subquestions they felt needed to be answered in order to assemble a satisfactory answer to the question. (For ease of reference, we will refer to this set of question decompositions as $QD_{human}$.) The subquestions generated by the annotators were then compiled into a "pyramid" structure similar to the ones proposed in (Nenkova and Passonneau, 2004). In order to create pyramids, humans first identified subquestions that sought the same information (or were reasonable paraphrases of each other) and then assigned each unique question a score equal to the number of times it appeared in the question decompositions produced by all annotators. Second, we utilized our random walk model to generate a set of question decompositions ($QD_{auto1}$) for each complex questions. Third, as shown in Figure 1, the subquestions in $QD_{auto1}$ were used to generate multi-document summaries which were used to automatically generate a fourth set of question decompositions ($QD_{auto2}$). As with $QD_{human}$, the subquestions generated for $QD_{auto1}$ and $QD_{auto2}$ were combined into pyramid structures by human annotators.

Each of these three sets of question decompositions were then compared against a set of "gold standard" decompositions created by another team of 4 human annotators from from the 4 "model summaries" prepared by NIST annotators as "gold standard" answers to the 8 complex questions. Each of the three question decompositions described above (i.e. $QD_{human}$, $QD_{auto1}$, and $QD_{auto2}$) were then scored against the corresponding "model" question decomposition pyramid using the technique outlined in [14]. Table 6 illustrates the Pyramid coverage for $QD_{auto1}$, $QD_{auto2}$, and $QD_{human}$ It is to be noted that although the $QD_{human}$ captured 45% of the questions contained in the "model" pyramids, the high average Pyramid score (0.5000) suggests that human question decompositions typically included questions that corresponded to the most vital information identified by the authors of the "model" summaries.

Another important observation is that the coverage and the Pyramid score of $QD_{auto2}$ are almost 80% of the same measures obtained for $QD_{human}$, whereas the Pyramid score of the question decompositions $QD_{auto1}$ is only 45% of the Pyramid score and coverage obtained for $QD_{human}$. In fact, these scores vary based on the number of feedback loops allowed for the Answer Summary 3 from Figure 1. Figure 5 illustrates the average average Pyramid scores that were obtained at each step of the feedback loop for all eight questions, both for $QD_{auto1}$, and $QD_{auto2}$. The Figure

| Topic<br>Description | Pyramid Score for Question Decompositions | | |
|---|---|---|---|
| | $QD_{auto1}$ | $QD_{auto2}$ | $QD_{human}$ |
| Falkland Islands | 0.2012 | 0.3202 | 0.3889 |
| Tourist Attacks | 0.2317 | 0.3745 | 0.5000 |
| Drug Development | 0.3114 | 0.5195 | 0.6744 |
| Amazon Rainforest | 0.2500 | 0.4091 | 0.6000 |
| Welsh Government | 0.2931 | 0.4873 | 0.5091 |
| Robot Technology | 0.2268 | 0.4421 | 0.6222 |
| U.K. Tourism | 0.0196 | 0.3917 | 0.4035 |
| Czechoslovakia | 0.2301 | 0.3116 | 0.3836 |
| AVERAGE | 0.2205 | 0.4070 | 0.5000 |

**Table 6: Pyramid Coverage of Question Decompositions.**

shows that the Pyramid scores improve for $QD_{auto1}$. The improvement for $QD_{auto2}$ is less dramatic. This means that the comparison and selection of new question decompositions at each feedback loop determines better questions and better answers.

| Topic<br>Description | Responsiveness Score | | | |
|---|---|---|---|---|
| | Summary 1 | Summary 2 | Summary 3 | Human Sum |
| Falkland Islands | 3.75 | 4.00 | 4.00 | 4.50 |
| Tourist Attacks | 2.75 | 3.00 | 3.25 | 4.75 |
| Drug Development | 2.00 | 2.75 | 3.00 | 4.50 |
| Amazon Rainforest | 3.00 | 3.25 | 4.00 | 4.75 |
| Welsh Government | 3.75 | 4.00 | 3.75 | 5.00 |
| Robot Technology | 3.00 | 3.50 | 4.00 | 4.50 |
| U.K. Tourism | 3.75 | 4.00 | 4.25 | 4.25 |
| Czechoslovakia | 2.25 | 3.00 | 4.00 | 4.50 |
| AVERAGE | 3.03 | 3.44 | 3.72 | 4.59 |

**Table 7: Responsiveness Score for the Human Summaries and Answer Summaries 1, 2 and 3.**

Four different similarity metrics are responsible for the comparisons. They are listed in Figure 4. Pairs of these similarity metrics were also used for defining the relevance of question decompositions to each complex question. The aggregate similarity between $q_i \in QD_{auto1}$ and $QD_{auto2}$ is defined as $A\text{-}sim(q_i, QD_{auto2}) = \frac{1}{7}\sum_{j=1}^{7} sim_j(q_i, QD_{auto2})$. The similarity scores play an important role in the selections of questions from $QD_{auto1}$ for the next loop. In our experiments, we observed that if we take only similarity 3 we obtain the best results.

**Similarity Metric 1** weights content terms in $QD_{auto1}$ and $QD_{auto2}$ using $tfidf$ ($w_i = w(t_i) = (1 + \log(tf_i))\frac{\log N}{df_i}$), where $N$ is the number of questions in $QD_{auto1}$ and $QD_{auto2}$, while $df_i$ is equal to the number of questions in containing $t_i$ and $tf_i$ is the number of times $t_i$ appears in $QD_{auto1}$ and $QD_{auto2}$. Any question in $QD_{auto1}$ and $QD_{auto2}$ can be transformed into two vectors, $v_q = \langle w_{q_1}, w_{q_2}, ..., w_{q_m} \rangle$ and $v_u = \langle w_{u_1}, w_{u_2}, ..., w_{u_n} \rangle$; The similarity between $QD_{auto1}$ and $QD_{auto2}$ is measured as the cosine measure between their corresponding vectors:: $\cos(v_q, v_u) = (\sum_i w_{q_i} w_{u_i})/((\sum_i w_{q_i}^2)^{\frac{1}{2}} \times (\sum_i w_{u_i}^2)^{\frac{1}{2}})$.

**Similarity Metric 2** is based on the percent of terms in $QD_{auto1}$ that appear in the $QD_{auto2}$. It is obtained by finding the intersection of the terms in the term vectors of the two questions.

**Similarity Metric 3** utilizes semantic information from WordNet. It involves:
(a) finding the minimum path between WordNet concepts. Given two terms $t_1$ and $t_2$, each with $n$ and $m$ WordNet senses $S_1 = \{s_1, ..., s_n\}$ and $S_2 = \{r_1, ..., r_m\}$. The semantic distance between the terms $\delta(t_1, t_2)$ is defined by the minimum of all the possible pair-wise semantic distances between $S_1$ and $S_2$: $\delta(t_1, t_2) = \min_{s_i \in S_1, r_j \in S_2} D(s_i, r_j)$, where $D(s_i, r_j)$ is the path length between $s_i$ and $r_j$.
(b) The semantic similarity between the vector transformations $v_q$ and $v_u$ from $QD_{auto1}$ and $QD_{auto2}$ respectively is defined as $sem(v_q, v_u) = \frac{I(v_q, v_u) + I(v_u, v_q)}{|v_q| + |v_u|}$, where $I(v_x, v_y) = \sum_{x \in v_x} \frac{1}{1 + \min_{y \in v_y} \delta(x, y)}$

**Similarity Metric 4** is based on question type similarity, using a question-type similarity matrix similar to the one introduced in [11].

**Figure 4: The four similarity metrics.**

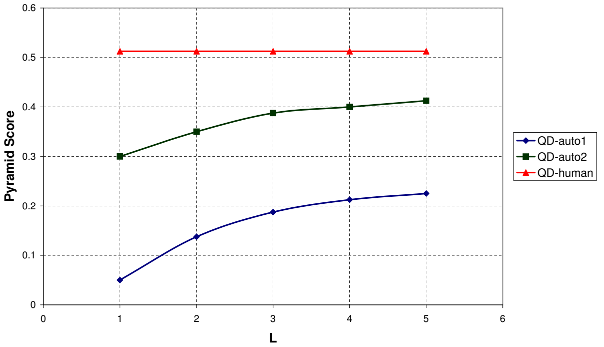

**Figure 5: Pyramid scores at each step of the feedback loop.**

**Evaluation of Answers.**

Answers were evaluated by the "responsiveness score" designed by the NIST assessors. The score provides a coarse ranking of the summaries for each topic, according to the amount of information in the summary that helps to satisfy the information need expressed in the topic statement. Four linguist assigned these scores for all three forms of answer summaries. Table 7 illustrates the responsiveness scores for Answer Summary 1, Answer Summary 2, Answer Summary 3, from Figure 1 and the human generated summaries. The responsiveness score is measured on a scale from 1 to 5 and it quantifies how well does a summary answer the complex question. A score of 1 is the least responsive to the question. A score of 5 means that the summary answers completely the question.

**Evaluation of the Impact of the Decomposed Questions on Answer Summaries.**

We were also interested to evaluate the impact the question decompositions would have when we select different values for the parameter $\alpha$ which stops the Markov chain. Figure 6 illustrates the average Responsiveness score obtained when $\alpha = 0.85$, $\alpha = 0.6$, $\alpha = 0.5$, $\alpha = 0.3$ and $\alpha = 0.15$. Since the question decompositions determine two different answers, as it was illustrated in Figure 1, we have measured the responsiveness for both of them and illustrate the results in Figure 6.

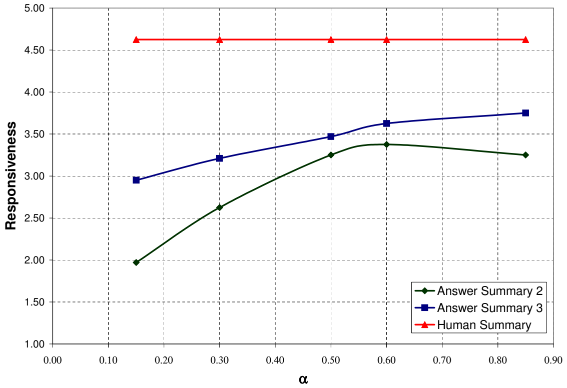

**Figure 6: Responsiveness for different $\alpha$ values.**

In a separate effort, we evaluated the impact of only the question decompositions that were considered relevant to the complex question by the random walk presented in Section 4.2. Since that random walk depends on the parameter $d$, we have tested the question coverage for $d = 0.85$, $d = 0.6$, $d = 0.5$, $d = 0.3$, and $d = 0.15$. Figure 7 illustrates the average Responsiveness obtained in this case. Since only Answer Summary 3 is obtained by considering the relevance of question decompositions to the complex question, unlike Figure 6, in Figure 7 we illustrate results only for Answer Summary 3. The best results are obtained for $d = 0.85$. This result supports our intuition that the question decompositions should not be necessarily very different, but they must be relevant to the complex question. The difference from the responsiveness of human-generated summaries indicates that relevance takes into account more sophisticated information than the one contained in questions.

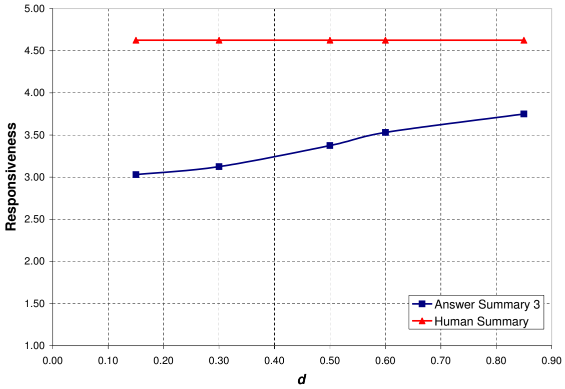

**Figure 7: Responsiveness for different $d$ values.**

## 6. CONCLUSIONS

We have presented a new framework for question decomposition that allows several forms of answers to be returned for complex questions. Two forms of random walks were used. The first random walk was used for surfing the space of relations relevant to the complex question, in order to generate question decompositions. The second random walk was used for measuring the relevance of the question decompositions to the complex question.

The evaluations have shown that the question decompositions lead to more relevant and complete answers. The results have also shown that the coverage of automatically generated question decompositions, when compared with the questions generated from the answer summary are better indicators of answer quality than the relevance score to the complex question. The evaluations have also indicated the question coverage for automatic methods is 85% of the coverage of questions produced by humans.

In this paper we have also described a Q/A architecture which allows feedback loops for improving the quality of answers through the coverage of question decompositions.

## 7. ACKNOWLEDGMENTS

This material is based upon work funded in whole or in part by the U.S. Government and any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the U.S. Government.

## 8. REFERENCES

[1] E. Brill. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging. *Computational Linguistics*, 21(4), 1995.

[2] M. Collins. *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, 1999.

[3] S. Harabagiu. Incremental Topic Representations. In *Proceedings of the 20th COLING Conference*, Geneva, Switzerland, 2004.

[4] S. Harabagiu, A. Hickl, J. Lehmann, and D. Moldovan. Experiments with Interactive Question-Answering. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, 2005.

[5] S. Harabagiu, D. Moldovan, C. Clark, M. Bowden, A. Hickl, and P. Wang. Employing Two Question Answering Systems in TREC 2005. In *Proceedings of the Fourteenth Text REtrieval Conference*, 2005.

[6] S. Harabagiu, D. Moldovan, C. Clark, M. Bowden, J. Williams, and J. Bensley. Answer Mining by Combining Extraction Techniques with Abductive Reasoning. In *Proceedings of the Twelfth Text REtrieval Conference*, 2003.

[7] F. Lacatusu, A. Hickl, P. Aarseth, and L. Taylor. Lite-GISTexter at DUC 2005. In *Proceedings of the Document Understanding Workshop (DUC-2005)* , 2005.

[8] F. Lacatusu, A. Hickl, and S. Harabagiu. Impact of Question Decomposition on the Quality of Answer Summaries. In *Proceedings of the fifth international conference on Language Resources and Evaluation, (LREC 2006)*, 2006.

[9] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *2001 ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2001.

[10] C.-Y. Lin and E. Hovy. The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th COLING Conference*, Saarbrücken, Germany, 2000.

[11] S. Lytinen and N. Tomuro. The Use of Question Types to Match Questions in FAQFinder. In *Papers from the 2002 AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases*, pages 46–53, 2002.

[12] G. A. Miller. WordNet: a lexical database for English. *Communications of the Association for Computing Machinery*, 38(11):39–41, 1995.

[13] S. Narayanan and S. Harabagiu. Question Answering based on Semantic Structures. In *Proceedings of COLING-2004*, 2004.

[14] A. Nenkova and R. Passonneau. Evaluating Content Selection in Summarization: the Pyramid Method. In *HLT-NAACL 2004*, Boston, MA, 2004.

[15] J. Otterbacher, G. Erkan, and D. Radev. Using random walks for question-focused sentence retrieval. In *Proceedings of Human Language Technology Conference and Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*, Vancouver, Canada, 2005.

[16] M. Palmer, D. Gildea, and P. Kingsbury. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106, 2005.

[17] M. Pasca and S. Harabagiu. High Performance Question/Answering. In *Proceedings of the 24th Annual International ACM SIGIR Conference*, 2001.

[18] M. Thelen and E. Riloff. A Bootstrapping Method for Learning Semantic Lexicons using Extraction Pattern Contexts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, 2002.