Towards Topic-to-Question Generation

Yllias Chali*
University of Lethbridge

Sadid A. Hasan**
Philips Research North America

This paper is concerned with automatic generation of all possible questions from a topic of interest. Specifically, we consider that each topic is associated with a body of texts containing useful information about the topic. Then, questions are generated by exploiting the named entity information and the predicate argument structures of the sentences present in the body of texts. The importance of the generated questions is measured using Latent Dirichlet Allocation by identifying the subtopics (which are closely related to the original topic) in the given body of texts and applying the Extended String Subsequence Kernel to calculate their similarity with the questions. We also propose the use of syntactic tree kernels for the automatic judgment of the syntactic correctness of the questions. The questions are ranked by considering both their importance (in the context of the given body of texts) and syntactic correctness. To the best of our knowledge, no previous study has accomplished this task in our setting. A series of experiments demonstrate that the proposed topic-to-question generation approach can significantly outperform the state-of-the-art results.

1. Introduction

We live in an information age where all kinds of information is easily accessible through the Internet. The increasing demand for access to different types of information available online have interested researchers in a broad range of Information Retrieval–related areas, such as question answering, topic detection and tracking, summarization, multimedia retrieval, chemical and biological informatics, text structuring, and text mining. Although search engines do a remarkable job in searching through a heap of information, they have certain limitations, as they cannot satisfy the end users' information need to have more direct access to relevant documents. For example, if we ask for the impact of the current global financial crisis in different parts of the world, we can expect to sift through thousands of results for the answer. This fact can be more understandable by the following scenario. When a user enters a query, they are served with a ranked list of relevant documents by the standard document retrieval systems (i.e., search engines),

* University of Lethbridge, 4401 University Drive West, Lethbridge, Alberta, T1K 3M4, Canada. E-mail: chali@cs.uleth.ca.
** Philips Research North America, 345 Scarborough Rd, Briarcliff Manor, New York, 10510, USA. E-mail: sadid.hasan@philips.com.

Submission received: 19 May, 2013; revised submission received: 26 May, 2014; accepted for publication: 22 June, 2014.

doi:10.1162/COLI_a_00206

and their search task is usually not over (Chali, Joty, and Hasan 2009). The next step for the user is to look into the documents themselves and search for the precise piece of information they were looking for. This method is time-consuming, and a correct answer could easily be missed by either an incorrect query resulting in missing documents or by careless reading. This is why Question Answering (QA) has received immense attention from the information retrieval, information extraction, machine learning, and natural language processing communities in the last 15 years (Hirschman and Gaizauskas 2001; Strzalkowski and Harabagiu 2008; Kotov and Zhai 2010).

The main goal of QA systems is to retrieve relevant answers to natural language questions from a collection of documents rather than using keyword matching techniques to extract documents. Automated QA research focuses on how to respond with exact answers to a wide variety of questions, including: factoid, list, definition, how, why, hypothetical, semantically constrained, and crosslingual questions (Simmons 1965; Kupiec 1993; Voorhees 1999; Hirschman and Gaizauskas 2001; Greenwood 2005; Wang 2006; Moldovan, Clark, and Bowden 2007). One of the main requirements of a QA system is that it must receive a well-formed question as input in order to come up with the best possible correct answer as output. Available studies revealed that humans are not very skilled in asking good questions about a topic of their interest. They are forgetful in nature; this often restricts them to properly express whatever that is peeking in their mind. Therefore, they would benefit from automated Question Generation (QG) systems that can assist in meeting their inquiry needs (Lauer, Peacock, and Graesser 1992; Graesser et al. 2001; Rus and Graesser 2009; Ali, Chali, and Hasan 2010; Kotov and Zhai 2010; Olney, Graesser, and Person 2012). Another benefit of QG is that it can be a good tool to help improve the quality of the QA systems (Graesser et al. 2001; Rus and Graesser 2009). These benefits of a QG system motivate us to address the important problem of topic-to-question generation, where the main goal is to generate all possible questions about a given topic. For example, given the topic *Apple Inc. Logos*, we would like to generate questions such as *What is Apple Inc.?*, *Where is Apple Inc. located?*, *Who designed Apple's Logo?*, and so forth.

The problem of **topic-to-question** generation can be viewed as a generalization of the problem of answering complex questions. Complex questions are essentially broader information requests about a certain topic, whose answers could be obtained from pieces of information scattered in multiple documents. For example, consider the complex question:[1] *Describe steps taken and worldwide reaction prior to the introduction of the Euro on January 1, 1999. Include predictions and expectations reported in the press.* This question is requesting an elaboration about the topic "Introduction of the Euro," which can be answered by following complex procedures such as question decomposition or inferencing and synthesizing information from multiple documents (e.g., multi-document summarization). Answering complex questions is not easy as it is not always understandable to which direction one should move to search for the answer to a complex question. This situation arises because of the wider focus of the topic that is inherent in the complex question in consideration. For example, a complex question like *Describe the tsunami disaster in Japan* has a wider focus without a single or well-defined information need. To narrow down the focus, this question can be decomposed into a series of simple questions such as *How many people were killed in the tsunami?*, *How many people became homeless?*, *Which cities were mostly damaged?*, and so on.

1 The example complex questions have been provided according to the guidelines of the Document Understanding Conference (DUC, `http://duc.nist.gov/`) (2005–2007) tasks.

Decomposing a complex question automatically into simpler questions in this manner such that each of them can be answered individually by using the state-of-the-art QA systems, and then combining the individual answers to form a single answer to the original complex question, has proven effective to deal with the complex question answering problem (Harabagiu, Lacatusu, and Hickl 2006; Hickl et al. 2006; Chali, Hasan, and Imam 2012). Moreover, the generated simple questions can be used as the list of important aspects to act as a guide[2] for selecting the most relevant sentences in producing more focused and more accurate summaries as the output of a summarization system (Chali, Hasan, and Imam 2011, 2012). From this discussion, it is obvious that the complex question decomposition problem can be generalized to the problem of topic-to-question generation to help improve the complex question answering systems.

In this article[3], we consider the task of automatically generating questions from topics and assume that each topic is associated with a body of texts having useful information about the topic. This assumption has been inherited from the process of how a human asks questions based on their knowledge. For example, if a person knows that a *university* is an educational institution, then they can ask a question about its faculty and students. In this research, our main goal is to generate fact-based questions[4] about a given topic from its associated content information. We generate questions by exploiting the named entity information and the predicate argument structures of the sentences (along with semantic roles) present in the given body of texts. The named entities and the semantic role labels are used to identify relevant parts of a sentence in order to form relevant questions about them. The importance of the generated questions is measured in two steps. In the first step, we identify whether the question is asking something about the topic or something that is very closely related to the topic. We call this the measure of **topic relevance**. For this purpose, we use Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan 2003) to identify the subtopics (which are closely related to the original topic) in the given body of texts and apply the Extended String Subsequence Kernel (ESSK) (Hirao et al. 2003) to calculate their similarity with the questions. In the second step, we judge the syntactic correctness of each generated question. We apply the tree kernel functions (Collins and Duffy 2001) and re-implement the syntactic tree kernel model according to Moschitti et al. (2007) for computing the syntactic similarity of each question with the associated content information. We rank the questions by considering their topic relevance and syntactic correctness scores. Experimental results show the effectiveness of our approach for automatically generating topical questions. The remainder of the article is organized as follows. Section 2 describes the related work. Section 3 presents the description of our QG system. Section 4 explains the experiments and shows evaluation results; Section 5 concludes.

2. Related Work

Recently, question generation has received immense attention from researchers and different methods have been proposed to accomplish the task in different relevant fields (Andrenucci and Sneiders 2005). McGough et al. (2001) proposed an approach to build a Web-based testing system with the facility of dynamic QG. Wang et al. (2008) showed

2 `http://www.nist.gov/tac/2011/Summarization/Guided-Summ.2011.guidelines.html`.

3 This article is a longer version of our previously published work (Chali and Hasan 2012c). We provide more theoretical descriptions and analyses, and conduct our experiments on a larger data set to report new results.

4 We mainly focus on generating *who*, *what*, *where*, *which*, *when*, *why*, and *how* questions in this research.

a method to automatically generate questions based on question templates (which are created from training on medical articles). Brown, Frishkoff, and Eskenazi (2005) described an approach to automatically generate questions to assess the user's vocabulary knowledge. Chen, Aist, and Mostow (2009) developed a method to generate questions automatically from informational text to mimic the reader's self-questioning strategy during reading. On the other hand, Agarwal, Shah, and Mannem (2011) considered the question generation problem beyond the sentence level and designed an approach that uses discourse connectives to generate questions from a given text. Several other QG models have been proposed over the years that deal with transforming answers to questions and utilizing question generation as an intermediate step in the question answering process (Echihabi and Marcu 2003; Hickl et al. 2005). There are some other researchers who have approached the task of generating questions for educational purposes (Mitkov and Ha 2003; Heilman and Smith 2010b).

Question asking and QG are important components in advanced learning technologies such as intelligent tutoring systems and inquiry-based environments (Graesser et al. 2001). A QG system is useful for building better question-asking facilities in intelligent tutoring systems. The Natural Language Processing (NLP), Natural Language Generation, Intelligent Tutoring System, and Information Retrieval communities have currently identified the Text-to-Question generation task as promising candidates for shared tasks[5] (Rus and Graesser 2009; Boyer and Piwek 2010). In the Text-to-Question generation task, a QG system is given a text, and the goal is to generate a set of questions for which the text contains answers. The task of generating a question about a given text can be typically decomposed into three subtasks. First, given the source text, a content selection step is necessary to select a target to ask about, such as the desired answer. Second, given a target answer, an appropriate question type is selected (i.e., the form of question to ask is determined). Third, given the content and question type, the actual question is constructed. Based on this principle, several approaches have been described in Boyer and Piwek (2010) that use named entity information, syntactic knowledge, and semantic structures of the sentences to perform the task of generating questions from sentences and paragraphs (Heilman and Smith 2010a; Mannem, Prasad, and Joshi 2010). Inspired by these works, we perform the task of topic-to-question generation using named entity information and semantic structures of the sentences. A task that is similar to ours is the task of keywords-to-question generation that has been addressed recently in Zheng et al. (2011). They propose a user model for jointly generating keywords and questions. However, their approach is based on generating question templates from existing questions, which requires a large set of English questions as training data. In recent years, some other related researchers have proposed the tasks of high-quality question generation (Ignatova, Bernhard, and Gurevych 2008) and generating questions from queries (Lin 2008). Fact-based question generation has been accomplished previously (Rus, Cai, and Graesser 2007; Heilman and Smith 2010b). We also focus on generating fact-based questions in this research.

Besides grammaticality, an effective QG system should focus deeply on the importance of the generated questions (Vanderwende 2008). This motivates the use of a question-ranking module in a typical QG system. Over-generated questions can be ranked using different approaches, such as statistical ranking methods, dependency parsing, identification of the presence of pronouns and named entities, and topic scoring (Heilman and Smith 2010a; Mannem, Prasad, and Joshi 2010; McConnell et al. 2011).

5 http://www.questiongeneration.org/QGSTEC2010.

However, most of these automatic ranking approaches ignore the aspects of complex paraphrasing by not considering lexical semantic variations (e.g., synonymy) when measuring the importance of the questions. In our work, we use LDA (Blei, Ng, and Jordan 2003) to identify the subtopics (which are closely related to the original topic) in the given body of texts. We choose LDA because in recent years it has become one of the most popular topic modeling techniques and has been shown to be effective in several text-related tasks, such as document classification, information retrieval, and question answering (Wei and Croft 2006; Misra, Cappé, and Yvon 2008; Celikyilmaz, Hakkani-Tur, and Tur 2010).

Once we have the subtopics, we apply ESSK (Hirao et al. 2003) to calculate their similarity with the generated questions. The choice of ESSK is motivated by its successful use in different NLP tasks in recent years (Chali, Hasan, and Joty 2009, 2011; Chali and Hasan 2012a, 2012b). Hirao et al. (2003) introduced ESSK considering all possible senses of each word to perform their summarization task. Their method is effective. However, the fact that they do not disambiguate word senses cannot be disregarded. In our task, we apply ESSK to calculate the similarity between important topics (discovered using LDA) and the generated questions in order to measure the importance of each question. We use disambiguated word senses for this purpose.

Syntactic information has previously been used successfully in question answering (Zhang and Lee 2003; Moschitti and Basili 2006; Moschitti et al. 2007; Chali, Hasan, and Joty 2009, 2011). Pasca and Harabagiu (2001) argued that with the syntactic form of a sentence one can see which words depend on other words. We also feel that there should be a similarity between the words that are dependent in the sentences present in the associated body of texts and the dependency between words of the generated question. This motivates us to propose the use of syntactic kernels in judging the syntactic correctness of the generated questions automatically.

The main goal of our work is to generate as many questions as possible related to the topic. We use the named entity information and the predicate argument structures of the sentences to accomplish this goal. Our approach is different from the set-up in shared tasks (Rus and Graesser 2009; Boyer and Piwek 2010), as we generate a set of basic questions that are useful to add variety in the question space. A paragraph associated with each topic is used as the source of relevant information about the topic. We evaluate our systems in terms of topic relevance, which is different from prior research (Heilman and Smith 2010a; Mannem, Prasad, and Joshi 2010). Syntactic correctness is also an important property of a good question. For this reason, we evaluate our system in terms of syntactic correctness as well. The proposed system will be useful for generating topic-related questions from the associated content information, which can be used to incorporate a "question suggestions for a certain topic" facility in search systems (Kotov and Zhai 2010). For example, if a user searches for some information related to a certain topic, the search system could generate all possible topic-relevant questions from a pre-existent related body of texts to provide suggestions. Kotov and Zhai (2010) approached a similar task by proposing a technique to augment the standard ranked list presentation of search results with a question-based interface to refine user-given queries.

The major contributions of our work can be summarized as follows:

- We perform the task of topic-to-question generation, which can help users in expressing their information needs. Questions are generated using a set of general-purpose rules based on named entity information and the predicate argument structures of the sentences (along with semantic roles) present in the associated body of texts.

- We identify the subtopics (which are closely related to the original topic) in the given body of texts by using LDA and calculate their similarity with the questions by applying ESSK (with disambiguated word senses). This helps us to measure the importance of each question.
- We compute the syntactic similarity of each question with its associated content information by applying the tree kernel functions with the re-implementation of the syntactic tree kernel model. In this way, we judge the syntactic correctness of each generated question automatically.
- We evaluate the ESSK similarity scores and the syntactic similarity scores in a ranking framework and show that the use of ESSK and syntactic kernels improve the relevance and the syntactic correctness of the top-ranked questions, respectively.
- We identify circumstances in which our approach performs well and show that, using additional experiments by narrowing down the topic focus. Experiments with the topics about people (biographical focus) reveal improvements in the overall results.

3. Topic-to-Question Generation

Our QG approach mainly builds on four steps. In the first step, complex sentences (from the given body of texts) related to a topic are simplified, as it is easier to generate questions from simple sentences. In the next step, named entity information and predicate argument structures of the sentences are extracted and are then used to generate questions. In the third step, LDA is used to identify important subtopics from the given body of texts, and then ESSK is applied to find their similarity with the generated questions. In the final step, a syntactic tree kernel is used and syntactic similarity between the generated questions and the sentences present in the body of texts determines the syntactic correctness of the questions. Questions are then ranked by considering the ESSK similarity scores and the syntactic similarity scores. We present an architectural diagram (Figure 1) to show the different components of our system and describe the overall procedure in the following subsections.

3.1 Sentence Simplification

Sentences may have complex grammatical structure with multiple embedded clauses. Therefore, the first step of our proposed system is to simplify the complex sentences with the intention of generating more accurate questions. We use the simplified factual statement extractor model[6] of Heilman and Smith (2010a). Their model extracts the simpler forms of the complex source sentence by altering lexical items, syntactic structure, and semantics, as well as by removing phrase types such as leading conjunctions, sentence-level modifying phrases, and appositives. For example, given a complex sentence *s*, we get the corresponding simple sentences as follows:

Complex Sentence (s): Apple's first logo, designed by Jobs and Wayne, depicts Sir Isaac Newton sitting under an apple tree.
Simple Sentence (1): Apple's first logo is designed by Jobs and Wayne.

6 Available at `http://www.ark.cs.cmu.edu/mheilman/questions/`.

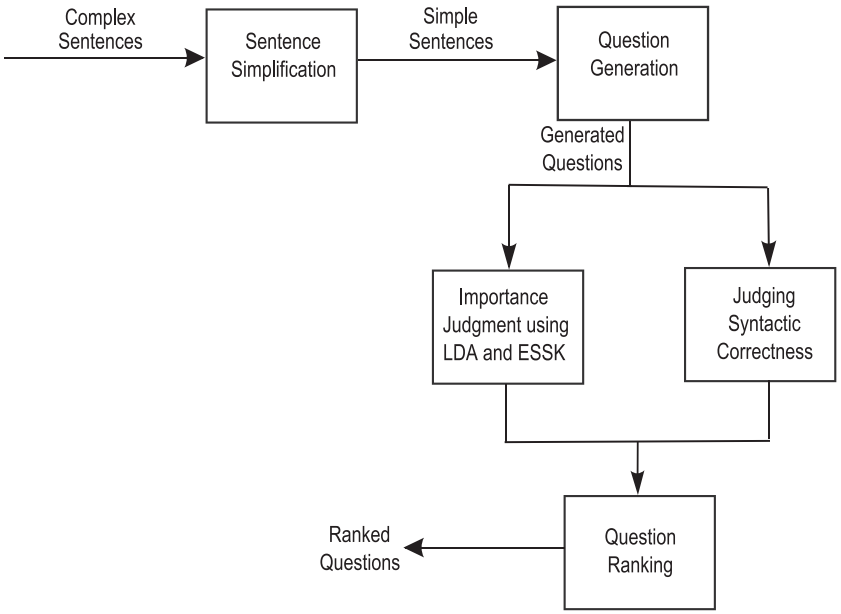

Figure 1
Architectural diagram of our system.

Simple Sentence (2): Apple's first logo depicts Sir Isaac Newton sitting under an apple tree.

3.2 Named Entity Information and Semantic Role Labeling for QG

In the second step of our system, we at first process the simple sentences in order to generate all possible questions from them. We use the Illinois Named Entity Tagger,[7] a state-of-the-art named entity (NE) tagger that tags plain text with named entities (people, organizations, locations, miscellaneous) (Ratinov and Roth 2009). Once we tag the topic under consideration and its associated body of texts, we use some general purpose rules to create some basic questions even though the answer is not present in the body of texts. For example, *Apple Inc.* is tagged as an organization, so we generate a question: *Where is Apple Inc. located?*. The main motivation behind generating such questions is to add variety to the generated question space. The basic questions are useful when there is very little or no knowledge available for a certain topic in consideration. This assumption is inherited from the scenario in the real world where a human can ask questions having very limited background knowledge about the topic. For example, if a person knows nothing about a *university*, they can ask *What is it?* or, if they at least know that a *university* is an institution, then they can ask the question, *Where is it located?*. In Table 1, we show some example rules for the basic questions generated in this work.

Our next task is to generate specific questions from the sentences present in the given body of texts. For this purpose, we parse the sentences semantically using a

7 Available at `http://cogcomp.cs.illinois.edu/`.

Table 1
Example basic question rules.

Tag	Example Question
person	Who is *person*?
organization	Where is *organization* located?
location	Where is *location*?
misc.	What do you know about *misc.*?

Semantic Role Labeling (SRL) system (Kingsbury and Palmer 2002; Hacioglu et al. 2003), ASSERT.[8] ASSERT is an automatic statistical semantic role tagger that can annotate naturally occuring text with semantic arguments. When presented with a sentence, it performs a full syntactic analysis of the sentence, automatically identifies all the verb predicates in that sentence, extracts features for all constituents in the parse tree relative to the predicate, and identifies and tags the constituents with the appropriate semantic arguments. For example, the output of the SRL system for the sentence *Apple's first logo is designed by Jobs and Wayne* is: [ARG1 Apple 's first logo] is [TARGET designed] [ARG0 by Jobs and Wayne]. The output contains one verb (predicate) with its arguments (i.e., semantic roles). These arguments are used to generate specific questions from the sentences. For example, we can replace *[ARG1 ..]* with *What* and generate a question as: *What is designed by Jobs and Wayne?*. Similarly, *[ARG0 ..]* can be replaced and the question: *Who designed Apple's first logo?* can be generated. The semantic roles *ARG0...ARG5* are called **mandatory arguments**. There are some additional arguments or semantic roles that can be tagged by ASSERT. They are called **optional arguments** and they start with the prefix *ARGM*. These are defined by the annotation guidelines set in Palmer, Gildea, and Kingsbury (2005). A set of about 350 general-purpose rules are used to transform the semantic-role labeled sentences into the questions. The rules were set up in a way that we could use the semantic role information to find the potential answer words in a sentence that would be replaced by suitable question words. In the case of a mandatory argument, the choice of question word depends on the argument's named entity tag (*who* for a person, *where* for a location, etc.). Table 2 shows how different semantic roles can be replaced by possible question words in order to generate a question.

3.3 Importance of Generated Questions

In the third step of our proposed system, we pass the generated questions to the importance judgment module that uses LDA and ESSK to assign a topic relevance score to each question. The detailed procedure is discussed in the following subsections.

3.3.1 Latent Dirichlet Allocation (LDA). To measure the importance of the generated questions, we use LDA (Blei, Ng, and Jordan 2003) to identify the important subtopics[9] from the given body of texts. LDA is a probabilistic topic modeling technique where the main principle is to view each document as a mixture of various topics. Here each topic is a probability distribution over words. LDA assumes that documents are made

8 Available at `http://cemantix.org/assert.html`.
9 The term *sub-topic* is used in the LDA topic modeling sense, which represents a probability distribution over words.

Table 2
Semantic roles with possible question words.

Arguments	Question Words
ARG0...ARG5	who, where, what, which
ARGM-ADV	in what circumstances
ARGM-CAU	why
ARGM-DIS	how
AGRM-EXT	to what extent
ARGM-LOC	where
ARGM-MNR	how
ARGM-PNC	why
ARGM-TMP	when

up of words and word ordering is not important ("bag-of-words" assumption) (Misra, Cappé, and Yvon 2008). The main idea is to choose a distribution over topics while generating a new document. For each word in the new document, a topic is randomly chosen according to this distribution and a word is drawn from that topic. LDA uses a generative topic modeling approach to specify the following distribution over words within a document:

$$P(w_i) = \sum_{j=1}^{K} P(w_i|z_i = j)P(z_i = j) \tag{1}$$

where K is the number of topics, $P(w_i|z_i = j)$ is the probability of word w_i under topic j, and $P(z_i = j)$ is the sampling probability of topic j for the *ith* word. The multinomial distributions $\phi^{(j)} = P(w|z_i = j)$ and $\theta^{(d)} = P(z)$ are termed as topic-word distribution and document-topic distribution, respectively (Blei, Ng, and Jordan 2003). A Dirichlet (α) prior is placed on θ and a Dirichlet (β) prior is set on ϕ to refine this basic model (Griffiths and Steyvers 2002; Blei, Ng, and Jordan 2003). Now the main goal is to estimate the two parameters: θ and ϕ. We apply this framework directly to solve our problem by considering each topic-related body of texts as a document. We use a GUI-based toolkit for topic modeling[10] that uses the popular MALLET (McCallum 2002) toolkit for the back-end. The LDA model is built on the development set[11] (Section 4.2). The process starts by removing a list of "stop words" from the document and runs 200 iterations of Gibbs sampling (Geman and Geman 1984) to estimate the parameters θ and ϕ. From each body of texts, we discover K topics and choose the most frequent words from the most likely unigrams as the desired subtopics. For example, from the associated body of texts of the topic *Apple Inc. Logos*, we get these subtopics: *janoff, themes, logo, color, apple*.

3.3.2 Extended String Subsequence Kernel (ESSK). Once we identify the subtopics, we apply ESSK (Hirao et al. 2003) to measure their similarity with the generated questions. In the general ESSK, each word in a sentence is considered an "alphabet," and the alternative is all its possible senses. However, our ESSK implementation considers the

10 Available at `http://code.google.com/p/topic-modeling-tool/`.
11 The model was built and tested according to the guidelines of the topic modeling toolkit we used.

alternative of each word as its disambiguated sense. We use a dictionary-based Word Sense Disambiguation (WSD) system (Chali and Joty 2007) assuming one sense per discourse. We use WordNet (Fellbaum 1998) to find the semantic relations (such as repetition, synonym, hypernym and hyponym, holonym and meronym, and gloss) for all the words in a text. We assign a weight to each semantic relation based on heuristics and use all of them. Our WSD technique is decomposed into two steps: (1) building a representation of all possible senses of the words and (2) disambiguating the words based on the highest score. To be specific, each candidate word from the context is expanded to all of its senses. A disambiguation graph is constructed as the intermediate representation where the nodes denote word instances with their WordNet senses, and the weighted edges (connecting the senses of two different words) represent semantic relations. This graph is exploited to perform the WSD. We sum the weights of all edges, leaving the nodes under their different senses. The sense with the highest score is considered to be the most probable sense. In case of a tie between two or more senses, we select the sense that comes first in WordNet, because WordNet orders the senses of a word by decreasing order of their frequency. Our preliminary experiments suggested that WSD has a positive impact on the performance of our proposed system.

ESSK is used to measure the similarity between all possible subsequences of the question words/senses and topic words/senses. We calculate the similarity score $\text{Sim}(T_i, Q_j)$ using ESSK, where T_i denotes a topic/sub-topic word sequence and Q_j stands for a generated question. Formally, ESSK is defined as follows:[12]

$$K_{essk}(T, Q) = \sum_{m=1}^{d} \sum_{t_i \in T} \sum_{q_j \in Q} K_m(t_i, q_j)$$

$$K_m(t_i, q_j) = \begin{cases} val(t_i, q_j) & \text{if } m = 1 \\ K'_{m-1}(t_i, q_j) \cdot val(t_i, q_j) \end{cases}$$

Here, $K'_m(t_i, q_j)$ is defined in the following. t_i and q_j are nodes of T and Q, respectively. The function $val(t, q)$ returns the number of common attributes (i.e., the number of common words/senses) to the given nodes t and q.

$$K'_m(t_i, q_j) = \begin{cases} 0 & \text{if } j = 1 \\ \lambda K'_m(t_i, q_{j-1}) + K''_m(t_i, q_{j-1}) \end{cases}$$

Here, λ is the decay parameter for the number of skipped words. $K''_m(t_i, q_j)$ is defined as:

$$K''_m(t_i, q_j) = \begin{cases} 0 & \text{if } i = 1 \\ \lambda K''_m(t_{i-1}, q_j) + K_m(t_{i-1}, q_j) \end{cases}$$

12 The formulae denote a dynamic programming technique to compute the ESSK similarity score where d is the vector space dimension (i.e., the number of all possible subsequences of up to length d). More information about these formulae can be obtained from Hirao et al. (2003, 2004).

Finally, the similarity measure is defined after normalization:

$$sim_{essk}(T, Q) = \frac{K_{essk}(T, Q)}{\sqrt{K_{essk}(T, T)K_{essk}(Q, Q)}}$$

3.4 Judging Syntactic Correctness

The next step of our system is to judge the syntactic correctness of the generated questions. The generated questions might be syntactically incorrect due to the process of automatic question generation. It is time-consuming and considerable human intervention is necessary to check for the syntactically incorrect questions manually. We strongly believe that a question should have a similar syntactic structure to a sentence from which it is generated. For example, the sentence *Apple's first logo is designed by Jobs and Wayne.*, and the generated question *What is designed by Jobs and Wayne?* are syntactically similar. An example of an ungrammatical generated question that is not very similar to its source is: *Janoff presented Jobs What?*. To judge the syntactic correctness of each generated question automatically, we apply the tree kernel functions and re-implement the syntactic tree kernel model according to Moschitti et al. (2007) for computing the syntactic similarity of each question with the associated content information. We first parse the sentences and the questions into syntactic trees using the Charniak parser[13] (Charniak 1999). Then, we calculate the similarity between the two corresponding trees using the *tree kernel* method (Collins and Duffy 2001). We convert each parenthetic representation generated by the Charniak parser into its corresponding tree and give the trees as input to the tree kernel functions for measuring the syntactic similarity.

The tree kernel function computes the number of common subtrees between two trees and gives the similarity score between each sentence in the given body of texts and the generated question based on the syntactic structure. Each sentence[14] contributes a score to the questions and then the questions are ranked by considering the average of similarity scores.

4. Experiments

4.1 System Description

We consider the task of automatically generating questions from topics where each topic is associated with a body of texts having a useful description about the topic. The question-ranking module of the proposed QG system ranks the questions by combining the topic relevance scores and the syntactic similarity scores of Section 3.3 and Section 3.4 using the following formula:

$$w * ESSK_{score} + (1 - w) * SYN_{score} \quad (2)$$

13 Available at `https://github.com/BLLIP/bllip-parser`.

14 We consider that a question is syntactically fluent as well as relevant to the topic if it has similar syntactic subtrees to those of the most sentences in the body of texts.

Here, w is the importance parameter, which holds a value in $[0, 1]$. We kept $w = 0.5$ to give equal importance[15] to topic relevance and syntactic correctness.

4.2 Corpus

To run our experiments, we use the data set provided in the Question Generation Shared Task and Evaluation Challenge[16] (2010) for the task of question generation from paragraphs. This data set consists of 60 paragraphs about 60 topics that were originally collected from several Wikipedia, OpenLearn, and Yahoo!Answers articles. The paragraphs contain around 5–7 sentences for a total of 100–200 tokens (including punctuation). This data set includes a diversity of topics of general interest. We consider these topics and treat the paragraphs as their associated useful content information in order to generate a set of questions using our proposed QG approach. We randomly select 10 topics and their associated paragraphs as the development data.[17] A total of 2,186 questions are generated from the remaining 50 topics (test data) to be ranked.

4.3 Evaluation Set-up

4.3.1 Methodology. We use a methodology derived from Boyer and Piwek (2010) and Heilman and Smith (2010b) to evaluate the performance of our QG systems. Three native English-speaking university graduate students judge the quality of the top-ranked 20% questions using two criteria: topic relevance and syntactic correctness. For topic relevance, the given score is an integer between 1 (very poor) and 5 (very good) and is guided by the consideration of the following aspects: 1. Semantic correctness (i.e., the question is meaningful and related to the topic), 2. Correctness of question type (i.e., a correct question word is used), and 3. Referential clarity (i.e., it is clearly possible to understand what the question refers to). For syntactic correctness, the assigned score is also an integer between 1 (very poor) and 5 (very good). Whether a question is grammatically correct or not is checked here. The judges were asked to read the topics with their associated body of texts and then rate the top-ranked questions generated by different systems. For each question, we calculate the average of the judges' scores. The judges were provided with an annotation guideline and sample judgments, according to the methodology derived from Boyer and Piwek (2010) and Heilman and Smith (2010b). The same judges evaluated all the system outputs and they were blind to the system identity when judging. No guidelines were provided on the relative importance of the various aspects that made the judgment task subjective. The inter-annotator agreement of Fleiss's $\kappa = 0.41, 0.45, 0.62$, and 0.33 are computed for the three judges for the results in Tables 3–6, indicating moderate (for the first two tables), and substantial and fair agreement (Landis and Koch 1977) between the raters, respectively. These κ values were shown to be acceptable in the literature for the relevant NLP tasks (Dolan and Brockett 2005; Glickman, Dagan, and Koppel 2005; Heilman and Smith 2010b).

15 A syntactically incorrect question is not useful even if it is relevant to the topic. This motivated us to give equal importance to topic relevance and syntactic correctness. The parameter w can be tuned to investigate its impact on the system performance.

16 `http://www.questiongeneration.org/mediawiki`.

17 We use these data to build necessary general purpose rules for our QG model.

Table 3
Topic relevance and syntactic correctness scores.

Systems	Topic Relevance	Syntactic Correctness
Baseline1 (No Ranking)	2.15	2.63
Baseline2 (Topic Signature)	3.24	3.30
State-of-the-art (Heilman and Smith 2010b)	3.35	3.45
Proposed QG System	3.48	3.55

4.3.2 Systems for Comparison. We report the performance of the following systems in order to do a meaningful comparison with our proposed QG system:

(1) Baseline1: This is our QG system without any question-ranking method applied to it. Here, we randomly select top 20% questions and rate them.
(2) Baseline2: For our second baseline, we build a QG system using an alternative topic modeling approach. Here, we use a topic signature model (instead of using LDA as discussed in Section 3.3.1) (Lin and Hovy 2000) to identify the important subtopics from the sentences present in the body of texts. The subtopics are the important words in the context that are closely related to the topic and have significantly greater probability of occurring in the given text compared with a large background corpus. We use a topic signature computation tool[18] for this purpose. The background corpus that is used in this tool contains 5,000 documents from the English GigaWord Corpus. For example, from the given body of texts of the topic *Apple Inc. Logos*, we get these subtopics: *jobs, logo, themes, rainbow, monochromatic*. Then we use the same steps of Sections 3.3.2 and 3.4, and use Equation (2) to combine the scores. We evaluate the top-ranked 20% questions and show the results.
(3) State-of-the-art: We choose a publicly available state-of-the-art QG system[19] to generate questions from the sentences in the body of texts. This system was shown to achieve good performance in generating fact-based questions about the content of a given article (Heilman and Smith 2010b). Their method ranks the questions automatically using a logistic regression model. Given a paragraph as input, this system processes each sentence and generates a set of ranked questions for the entire paragraph. We evaluate the top-ranked 20% questions[20] and report the results.

4.3.3 Results and Discussion. Table 3 shows the average topic relevance and syntactic correctness scores for all the systems. From these results, we can see that the *proposed QG* system improves the topic relevance and syntactic correctness scores over the *Baseline1* system by 62% and 35%, respectively, and improves the topic relevance and syntactic correctness scores over the *Baseline2* system by 7%, and 8%, respectively. On the other hand, the *proposed QG* system improves the topic relevance and syntactic correctness scores over the *state-of-the-art* system by 4% and 3%, respectively. From these results, we can clearly observe the effectiveness of our proposed QG system. The improvements in the results are statistically significant[21] ($p < 0.05$).

The main goal of this work was to generate as many questions as possible related to the topic. For this reason, we considered generating the basic questions. These questions

18 Available at `http://www.cis.upenn.edu/~lannie/topicS.html`.
19 Available at `http://www.ark.cs.cmu.edu/mheilman/questions/`.
20 We ignore the yes-no questions for our task.
21 We tested statistical significance using Student's t test.

Table 4
Acceptability of the questions (in %).

Systems	Top 15%	Top 30%
Baseline1 (No Ranking)	35.2	32.6
Baseline2 (Topic Signature)	45.9	33.8
State-of-the-art (Heilman and Smith 2010b)	44.7	38.5
Proposed QG System	46.5	40.6

Table 5
Topic relevance and syntactic correctness scores (narrowed focus).

Systems	Topic Relevance	Syntactic Correctness
Baseline1 (No Ranking)	2.84	2.75
Baseline2 (Topic Signature)	3.50	3.42
State-of-the-art (Heilman and Smith 2010b)	3.63	3.56
Proposed QG System	3.78	3.72

were also useful to provide variety in the question space. We generated these questions using the named entity information. As the performance of the NE taggers were unsatisfactory, we had a few of these questions generated. In most cases, these questions were outranked by other important questions, which included a combination of topics and subtopics to show higher topic relevance score measured by ESSK. Therefore, they do not have a considerable impact on the evaluation statistics. We claim that the overall performance of our systems could be further improved if the accuracy of the NE tagger and the semantic role labeler could be increased.

Acceptability Test. In another evaluation setting, the three annotators judge the questions for their overall acceptability as a good question. If a question shows no deficiency in terms of the criteria considered for topic relevance and syntactic correctness, it is termed as *acceptable*. We evaluate the top 15% and top 30% questions separately for each QG system and report the results indicating the percentage of questions rated as acceptable in Table 4. The results indicate that the percentage of the questions rated acceptable is reduced when we evaluate a greater number of questions—which proves the effectiveness of our QG system.

Narrowing Down the Focus. We run further experiments by narrowing down the topic focus. We consider only the topics about people (biographical focus). We choose 50 people as our topics from the list of the 20th century's 100 most influential people, published in *Time* magazine in 1999 and obtained the paragraphs containing their biographical information from Wikipedia articles.[22] We generate a total of 1,845 questions from the 50 topics considered and rank them using different ranking schemes as discussed before. We evaluate the top 20% questions using the similar evaluation methodologies and report the results in Table 5. From these results, we can see that the *proposed QG*

22 http://en.wikipedia.org/wiki/Time_100.

Table 6
Acceptability of the questions in % (narrowed focus).

Systems	Top 15%	Top 30%
Baseline1 (No Ranking)	38.6	31.5
Baseline2 (Topic Signature)	47.1	35.5
State-of-the-art (Heilman and Smith 2010b)	52.4	40.2
Proposed QG System	55.8	42.0

system improves the topic relevance and syntactic correctness scores over the *Baseline1* system by 33% and 35%, respectively, and improves the topic relevance and syntactic correctness scores over the *Baseline2* system by 8% and 9%, respectively. Moreover, the *proposed QG* system improves both the topic relevance and syntactic correctness scores over the *state-of-the-art* system by 4%. From these results, we can clearly observe the effectiveness of our proposed QG system when we narrow down the topic focus. We also evaluate the top 15% and top 30% questions separately for each QG system and report the results, indicating the percentage of questions rated as acceptable in Table 6. From these tables, we can clearly see the improvements in all the scores for all the QG approaches. This is reasonable because the accuracy of the NE tagger and the semantic role labeler is increased for the biographical data.[23] These results further demonstrate that the proposed system is significantly better (at $p < 0.05$) than the other considered systems. We plan to make our created resources available to other researchers.

4.3.4 An Input-Output Example. An input to our systems is, for instance,[24] the topic *Apple Inc. Logos* with the associated content information (body of texts):

> Apple's first logo, designed by Jobs and Wayne, depicts Sir Isaac Newton sitting under an apple tree. Almost immediately, though, this was replaced by Rob Janoff's "rainbow Apple", the now-familiar rainbow-colored silhouette of an apple with a bite taken out of it. Janoff presented Jobs with several different monochromatic themes for the "bitten" logo, and Jobs immediately took a liking to it. While Jobs liked the logo, he insisted it be in color to humanize the company. The Apple logo was designed with a bite so that it would be recognized as an apple rather than a cherry. The colored stripes were conceived to make the logo more accessible, and to represent the fact the monitor could reproduce images in color. In 1998, with the roll-out of the new iMac, Apple discontinued the rainbow theme and began to use monochromatic themes, nearly identical in shape to its previous rainbow incarnation.

The output of our systems is the ranked lists of questions. We show an example output in Table 7. To provide a more detailed analysis of our results, the average output scores of the example questions are presented in Table 8. From this table, we can understand how different aspects of the evaluation criteria affected the performance of the different systems. For example, Q1 of the proposed system was given a very good score due to

23 Although a few basic questions were generated compared with other important questions containing topical words, we believe they did not have a considerable impact on the overall performance of our system.

24 The example input text is provided from the Question Generation Shared Task and Evaluation Challenge (QGSTEC 2010) data set that we used for our experiments (Section 4.2).

Table 7
System output.

Systems	Top-ranked questions
Baseline2	Q1: Who presented Jobs with several different monochromatic themes for the bitten logo? Q2: What were conceived to make the logo more accessible? Q3: Who liked the logo?
State-of-the-art	Q1: Whose first logo depicts Sir Isaac Newton sitting under an apple tree? Q2: What depicts Sir Isaac Newton sitting under an apple tree? Q3: What did Janoff present Jobs with?
Proposed QG System	Q1: Who designed Apple's first logo? Q2: What was replaced by Rob Janoff's "rainbow Apple"? Q3: What were conceived to make the logo more accessible?

Table 8
Judgment scores associated with example questions.

Systems	Question	Average score
Baseline2	Q1	3.65
	Q2	3.42
	Q3	3.38
State-of-the-art	Q1	3.68
	Q2	3.63
	Q3	3.25
Proposed QG System	Q1	4.34
	Q2	3.50
	Q3	3.42

its relevance to the topic in consideration. On the other hand, Q3 of the state-of-the-art was assigned a lower score due to its lack of clarity with respect to the topic.

5. Conclusion

In this article, we have considered the task of automatically generating questions from topics where each topic is associated with a body of texts containing useful information. The proposed method exploits named entity and semantic role labeling information to accomplish the task. A key aspect of our approach was the use of latent Dirichlet allocation (LDA) to automatically discover the hidden subtopics from the sentences. We have proposed a novel method to rank the generated questions by considering: (1) subtopical similarity determined using ESSK algorithm in combination with word sense disambiguation, and (2) syntactic similarity determined using the syntactic tree kernel based method. We have compared the proposed question generation (QG) system with two baseline systems and one state-of-the-art system. The evaluation results show that the proposed QG system significantly outperforms all other considered systems, as our top-ranked system generated questions were found to be better in topic-relevance and syntactic correctness than those of the other systems. Our results demonstrated that judging syntactic correctness of the generated questions using the syntactic tree kernel based model was suitable in our question generation setting. We would like to further

our research by using other available measures such as an *n*-gram language model or a parser confidence score (Wagner, Foster, and van Genabith 2009) in order to see how they would perform on the same task. In this article, we have also extended our experiments by narrowing down the topic focus. In this experiment, we have considered *people* as topics. A rigorous analysis of the evaluation results has revealed that the performance of our proposed QG system can be enhanced if we narrow down the topic focus. We hope to carry on these ideas and develop further mechanisms for question generation based on the dependency features of the answers and answer finding (Li and Roth 2006; Pinchak and Lin 2006).

Acknowledgments

We would like to thank the anonymous reviewers for their useful comments. The research reported in this article was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada – discovery grant and the University of Lethbridge. This work was done when the second author was at the University of Lethbridge.

References

Agarwal, M., R. Shah, and P. Mannem. 2011. Automatic question generation using discourse cues. In *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–9, Portland, OR.

Ali, H., Y. Chali, and S. A. Hasan. 2010. Automation of question generation from sentences. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 58–67, Pittsburgh, PA.

Andrenucci, A. and E. Sneiders. 2005. Automated question answering: Review of the main approaches. In *Proceedings of the 3rd International Conference on Information Technology and Applications (ICITA'05)*, pages 514–519, Sydney.

Blei, D. M., A. Y. Ng, and M. I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Boyer, K. E. and P. Piwek, eds. 2010. *Proceedings of QG2010: The Third Workshop on Question Generation*. Pittsburgh, PA: `questiongeneration.org`.

Brown, J. C., G. A. Frishkoff, and M. Eskenazi. 2005. Automatic question generation for vocabulary assessment. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 819–826, Vancouver.

Celikyilmaz, A., D. Hakkani-Tur, and G. Tur. 2010. LDA based similarity modeling for question answering. In *Proceedings of the NAACL HLT 2010 Workshop on Semantic Search*, SS '10, pages 1–9, Los Angeles, CA.

Chali, Y. and S. A. Hasan. 2012a. On the effectiveness of using sentence compression models for query-focused multi-document summarization. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pages 457–474, Mumbai.

Chali, Y. and S. A. Hasan. 2012b. Query-focused multi-document summarization: Automatic data annotations and supervised learning approaches. *Journal of Natural Language Engineering*, 18(1):109–145.

Chali, Y. and S. A. Hasan. 2012c. Towards automatic topical question generation. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pages 475–492, Mumbai.

Chali, Y., S. A. Hasan, and K. Imam. 2011. An aspect-driven random walk model for topic-focused multi-document summarization. In *Proceedings of the 7th Asian Information Retrieval Societies Conference (AIRS 2011)*, pages 386–397, Dubai.

Chali, Y., S. A. Hasan, and K. Imam. 2012. Learning good decompositions of complex questions. In *Proceedings of the 17th International Conference on Applications of Natural Language Processing to Information Systems (NLDB 2012)*, pages 104–115, Groningen.

Chali, Y., S. A. Hasan, and S. R. Joty. 2009. Do automatic annotation techniques have any impact on supervised complex question answering? In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics (ACL-IJCNLP 2009)*, pages 329–332, Suntec.

Chali, Y., S. A. Hasan, and S. R. Joty. 2011. Improving graph-based random walks for complex question answering using syntactic, shallow semantic and

extended string subsequence kernels. *Information Processing & Management*, 47(6):843–855.
Chali, Y. and S. R. Joty. 2007. Word sense disambiguation using lexical cohesion. In *Proceedings of the 4th International Conference on Semantic Evaluations*, pages 476–479, Prague.
Chali, Y., S. R. Joty, and S. A. Hasan. 2009. Complex question answering: Unsupervised learning approaches and experiments. *Journal of Artificial Intelligence Research*, 35:1–47.
Charniak, E. 1999. A maximum-entropy-inspired parser. Technical Report CS-99-12, Brown University, Computer Science Department, Providence, RI.
Chen, W., G. Aist, and J. Mostow. 2009. Generating questions automatically from informational text. In *Proceedings of the 2nd Workshop on Question Generation (AIED 2009)*, pages 17–24, Arlington, VA.
Collins, M. and N. Duffy. 2001. Convolution Kernels for natural language. In *Proceedings of Neural Information Processing Systems*, pages 625–632, Vancouver.
Dolan, W. B. and C. Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the 3rd International Workshop on Paraphrasing (IWP2005)*, pages 9–16, Jeju Island.
Echihabi, A. and D. Marcu. 2003. A noisy-channel approach to question answering. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, pages 16–23, Sapporo.
Fellbaum, C. 1998. *WordNet - An Electronic Lexical Database*. MIT Press.
Geman, S. and D. Geman. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741.
Glickman, O., I. Dagan, and M. Koppel. 2005. A probabilistic classification approach for lexical textual entailment. In *AAAI*, pages 1,050–1,055, Pittsburgh, PA.
Graesser, A. C., K. VanLehn, C. P. Rose, P. W. Jordan, and D. Harter. 2001. Intelligent tutoring systems with conversational dialogue. *AI Magazine*, 22(4):39–52.
Greenwood, M. A. 2005. *Open-Domain Question Answering*. Ph.D. thesis, Department of Computer Science, University of Sheffield.
Griffiths, T. L. and M. Steyvers. 2002. Prediction and semantic association. In *NIPS'02*, pages 11–18, Cambridge, MA.
Hacioglu, K., S. Pradhan, W. Ward, J. H. Martin, and D. Jurafsky. 2003. Shallow semantic parsing using support vector machines. In Technical Report TR-CSLR-2003-03, University of Colorado, Boulder.
Harabagiu, S., F. Lacatusu, and A. Hickl. 2006. Answering complex questions with random walk models. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 220–227, Seattle, WA.
Heilman, M. and N. A. Smith. 2010a. Extracting simplified statements for factual question generation. In *Proceedings of the Third Workshop on Question Generation*, pages 11–20, Pittsburgh, PA.
Heilman, M. and N. A. Smith. 2010b. Good question! Statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617, Los Angeles, CA.
Hickl, A., J. Lehmann, D. Moldovan, and S. Harabagiu. 2005. Experiments with interactive question-answering. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 205–214, Ann Arbor, MI.
Hickl, A., P. Wang, J. Lehmann, and Sanda Harabagiu. 2006. Ferret: Interactive question-answering for real-world environments. In *Proceedings of the COLING/ACL on Interactive Presentation Sessions*, pages 25–28, Sydney.
Hirao, T., J. Suzuki, H. Isozaki, and E. Maeda. 2003. NTT's multiple document summarization system for DUC2003. In *Proceedings of the Document Understanding Conference*, Edmonton.
Hirao, T., J. Suzuki, H. Isozaki, and E. Maeda. 2004. Dependency-based sentence alignment for multiple document summarization. In *Proceedings of COLING 2004*, pages 446–452, Geneva.
Hirschman, L. and R. Gaizauskas. 2001. Natural language question answering: The view from here. *Natural Language Engineering*, 7(4):275–300.
Ignatova, K., D. Bernhard, and I. Gurevych. 2008. Generating high quality questions from low quality questions. In *Proceedings of the Workshop on the Question Generation Shared Task and Evaluation Challenge*, Arlington, VA.

Kingsbury, P. and M. Palmer. 2002. From Treebank to PropBank. In *Proceedings of the International Conference on Language Resources and Evaluation*, pages 1,989–1,993, Las Palmas.

Kotov, A. and C. Zhai. 2010. Towards natural question guided search. In *Proceedings of the 19th International Conference on the World Wide Web*, WWW '10, pages 541–550, Raleigh, NC.

Kupiec, J. 1993. MURAX: A robust linguistic approach for question answering using an on-line encyclopedia. In *SIGIR*, pages 181–190, Pittsburgh, PA.

Landis, J. R. and G. G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Lauer, T. W., E. Peacock, and A. C. Graesser, eds. 1992. *Questions and Information Systems*. Erlbaum, Hillsdale, NJ.

Li, X. and D. Roth. 2006. Learning question classifiers: The role of semantic information. *Journal of Natural Language Engineering*, 12(3):229–249.

Lin, C. Y. 2008. Automatic question generation from queries. In *Proceedings of the Workshop on the Question Generation Shared Task and Evaluation Challenge*, Arlington, VA.

Lin, C. Y. and E. H. Hovy. 2000. The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th Conference on Computational Linguistics*, pages 495–501, Saarbrüken.

Mannem, P., R. Prasad, and A. Joshi. 2010. Question generation from paragraphs at UPenn: QGSTEC system description. In *Proceedings of the Third Workshop on Question Generation*, pages 84–91, Pittsburgh, PA.

McCallum, A. K. 2002. MALLET: A machine learning for language toolkit. `http://mallet.cs.umass.edu`.

McConnell, C. C., P. Mannem, R. Prasad, and A. Joshi. 2011. A new approach to ranking over-generated questions. In *Proceedings of the AAAI Fall Symposium on Question Generation*, pages 45–48, Arlington, VA.

McGough, J., J. Mortensen, J. Johnson, and S. Fadali. 2001. A Web-based testing system with dynamic question generation. In *ASEE/IEEE Frontiers in Education Conference*, pages S3C-23–28 (vol. 3), Reno, NV.

Misra, H., O. Cappé, and F. Yvon. 2008. Using LDA to detect semantically incoherent documents. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, CoNLL '08, pages 41–48, Manchester.

Mitkov, R. and L. A. Ha. 2003. Computer-aided generation of multiple-choice tests. In *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing - Volume 2*, pages 17–22, Edmonton.

Moldovan, D., C. Clark, and M. Bowden. 2007. Lymba's PowerAnswer 4 in TREC 2007. In *Proceedings of the 16th Text REtreival Conference*, Gaithersburg, MD.

Moschitti, A. and R. Basili. 2006. A tree kernel approach to question and answer classification in question answering systems. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pages 1,510–1,513, Genoa.

Moschitti, A., S. Quarteroni, R. Basili, and S. Manandhar. 2007. Exploiting syntactic and shallow semantic kernels for question/answer classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 776–783, Prague.

Olney, A. M., A. C. Graesser, and N. K. Person. 2012. Question generation from concept maps. *Dialogue and Discourse*, 3(2):75–99.

Palmer, M., D. Gildea, and P. Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Pasca, M. and S. M. Harabagiu. 2001. Answer mining from on-line documents. In *Proceedings of the Association for Computational Linguistics 39th Annual Meeting and 10th Conference of the European Chapter Workshop on Open-Domain Question Answering*, pages 38–45, Toulouse.

Pinchak, C. and D. Lin. 2006. A probabilistic answer type model. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 393–400, Trento.

Ratinov, L. and D. Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155, Boulder, CO.

Rus, V., Z. Cai, and A. C. Graesser. 2007. Experiments on generating questions about facts. In *Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Text Processing*, pages 444–455, Mexico City.

Rus, V. and A. C. Graesser. 2009. The question generation shared task and evaluation challenge. In *Workshop on the Question Generation Shared Task and Evaluation Challenge, Final Report*, pages 1–37, University of Memphis.

Simmons, R. F. 1965. Answering English questions by computer: A survey. *Communications of the ACM*, 8(1):53–70.

Strzalkowski, T. and S. Harabagiu, 2008. *Advances in Open Domain Question Answering*. Springer.

Vanderwende, L. 2008. The importance of being important: Question generation. In *Proceedings of the Workshop on the Question Generation Shared Task and Evaluation Challenge*, Arlington, VA.

Voorhees, E. M. 1999. Overview of the TREC 1999 question answering track. In *Proceedings of the 8th Text REtreival Conference*, Gaithersburg, MD.

Wagner, J., J. Foster, and J. van Genabith. 2009. Judging grammaticality: Experiments in sentence classification. *CALICO Journal*, 26(3):474–490.

Wang, M. 2006. A survey of answer extraction techniques in factoid question answering. In *CMU 11-762 Language and Statistics II, literature review project*.

Wang, W., H. Tianyong, and L. Wenyin. 2008. Automatic question generation for learning evaluation in medicine. In *6th International Conference on Advances in Web Based Learning*, pages 242–251, Edinburgh.

Wei, X. and W. B. Croft. 2006. LDA-based document models for ad-hoc retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 178–185, Seattle, WA.

Zhang, A. and W. Lee. 2003. Question classification using support vector machines. In *Proceedings of the Special Interest Group on Information Retrieval*, pages 26–32, Toronto.

Zheng, Z., X. Si, E. Y. Chang, and X. Zhu. 2011. K2Q: Generating natural language questions from keywords with user refinements. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 947–955, Chiang Mai.