Towards Automatic Topical Question Generation

Yllias Chali *Sadid A. Hasan*
University of Lethbridge, Lethbridge, AB, Canada
chali@cs.uleth.ca, hasan@cs.uleth.ca

ABSTRACT
We address the challenge of automatically generating questions from topics. We consider that each topic is associated with a body of texts containing useful information about the topic. Questions are generated by exploiting the named entity information and the predicate argument structures of the sentences present in the body of texts. To measure the importance of the generated questions, we use Latent Dirichlet Allocation (LDA) to identify the sub-topics (which are closely related to the original topic) in the given body of texts and apply the Extended String Subsequence Kernel (ESSK) to calculate their similarity with the questions. We also propose the use of syntactic tree kernels for computing the syntactic correctness of the questions. The questions are ranked by considering their importance (in the context of the given body of texts) and syntactic correctness. To the best of our knowledge, no other study has accomplished this task in our setting before. Experiments show that our approach can significantly outperform the state-of-the-art results.

KEYWORDS: Question generation, named entity information, predicate argument structures, latent dirichlet allocation (LDA), extended string subsequence kernel (ESSK), syntactic tree kernel.

Proceedings of COLING 2012: Technical Papers, pages 475–492,
COLING 2012, Mumbai, December 2012.

1 Introduction

When a user is served with a ranked list of relevant documents by the standard document retrieval systems (i.e. search engines), his/her search task is usually not over (Chali et al., 2009b). The next step for him/her is to look into the documents themselves and search for the precise piece of information he/she was 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 (Kotov and Zhai, 2010). 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 which 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 (Olney et al., 2012; Ali et al., 2010; Kotov and Zhai, 2010; Rus and Graesser, 2009; Lauer et al., 1992; Graesser et al., 2001). Question asking and Question Generation are important components in advanced learning technologies such as intelligent tutoring systems, and inquiry-based environments (Graesser et al., 2001). A QG system would be useful for building better question asking facilities in intelligent tutoring systems. Another benefit of QG is that it can be a good tool to help improve the quality of the Question Answering (QA) systems (Graesser et al., 2001; Rus and Graesser, 2009).

The main motivation of this work is to generate all possible questions about a given topic. For example, given the topic *"Apple Inc. Logos"*, we can generate questions such as *"What is Apple Inc.?"*, *"Where is Apple Inc. located?"*, *"Who designed Apple's Logo?"* etc. We consider this 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. Our main goal is to generate fact-based questions[1] 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 over 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 et al., 2003) to identify the sub-topics (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 paper is organized as follows. Section 2 describes the related work and motivation followed by Section 3 that presents the description of our QG system. Section 4 explains the experiments and shows evaluation results. We conclude the paper in the next section.

[1] We mainly focus on generating *Who, What, Where, Which, When, Why and How* questions in this research.

2 Related Work and Motivation

Recently, question generation has got immense attention from the researchers and hence, 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 question generation. Wang et al. (2008) showed a method to automatically generate questions based on question templates (which are created from training on medical articles). Brown et al. (2005) described an approach to automatically generate questions to assess the user's vocabulary knowledge. To mimic the reader's self-questioning strategy during reading, Chen et al. (2009) developed a method to generate questions automatically from informational text. On the other hand, Agarwal et al. (2011) considered the question generation problem beyond sentence level and proposed 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).

The Natural Language Processing (NLP), Natural Language Generation (NLG), Intelligent Tutoring System, and Information Retrieval (IR) communities have currently identified the Text-to-Question generation task as promising candidates for shared tasks[2] (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 et al., 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 researches have proposed the tasks of high quality question generation (Ignatova et al., 2008) and generating questions from queries (Lin, 2008). Fact-based question generation has been accomplished previously by Rus et al. (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, identifying the presence of pronouns and

[2] http://www.questiongeneration.org/QGSTEC2010

named entities, and topic scoring (Heilman and Smith, 2010a; Mannem et al., 2010; McConnell et al., 2011). However, most of these automatic ranking approaches ignore the aspects of complex paraphrasing by not considering lexical semantic variations (e.g. synonymy) while measuring the importance of the questions. In our work, we use Latent Dirichlet Allocation (LDA) (Blei et al., 2003) to identify the sub-topics (which are closely related to the original topic) in the given body of texts. In recent years, LDA 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 (Misra et al., 2008; Wei and Croft, 2006; Celikyilmaz et al., 2010). Hirao et al. (2003) introduced ESSK considering all possible senses to 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 been used successfully in *question answering* previously (Chali et al., 2009a, 2011; Zhang and Lee, 2003; Moschitti et al., 2007; Moschitti and Basili, 2006). 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 which 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 NE information and the predicate argument structures of the sentences to accomplish this goal. Our approach is different from the setup in shared tasks (Rus and Graesser, 2009; Boyer and Piwek, 2010) as we generate a set of basic questions which 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 the prior works (Heilman and Smith, 2010a; Mannem et al., 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 to generate topic related questions from the associated content information which can be used to incorporate a "question suggestions for a certain topic" facility in the search systems. 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 preexistent 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 use LDA to identify the sub-topics (which are closely related to the original topic) in the given body of texts and apply ESSK (with disambiguated word senses) to calculate

their similarity with the questions. This helps us to measure the importance of each question.

- We apply the tree kernel functions and re-implement the syntactic tree kernel model for computing the syntactic similarity of each question with the associated content information. In this way, we judge the syntactic correctness of each generated question automatically.
- The ESSK similarity scores and the syntactic similarity scores are used to rank the generated questions. In doing so, we show that the use of ESSK and syntactic kernels improve the relevance and the syntactic correctness of the top-ranked questions, respectively.
- We also run experiments by narrowing down the topic focus. Experiments with the topics about persons (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 then, questions are generated using them. In the third step, LDA is used to identify important sub-topics from the given body of texts and then ESSK is applied to find their similarity with the generated questions. In the final step, syntactic tree kernel is employed 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 describe the overall procedure in the following subsections.

3.1 Sentence Simplification

Sentences may have complex grammatical structure with multiple embedded clauses. Therefore, we simplify the complex sentences with the intention to generate more accurate questions. We use the simplified factual statement extractor model[3] of Heilman and Smith (2010a). Their model extracts the simpler forms of the complex source sentence by altering lexical items, syntactic structure, and semantics and by removing phrase types such as leading conjunctions, sentence-level modifying phrases, and appositives. For example, given a complex sentence *s*, we get a corresponding simple sentence 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: Apple's first logo is designed by Jobs and Wayne.

3.2 Named Entity (NE) Information and Semantic Role Labeling (SRL) for QG

We use the Illinois Named Entity Tagger[4], a state of the art NE tagger that tags a plain text with named entities (people, organizations, locations, miscellaneous) (Ratinov and Roth, 2009).

[3] Available at http://www.ark.cs.cmu.edu/mheilman/
[4] Available at http://cogcomp.cs.illinois.edu/

Once we tag the topic in 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. Table 1 shows some example rules for basic questions generated in this work.

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

Table 1: Example basic question rules

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 Semantic Role Labeling (SRL) system (Kingsbury and Palmer, 2002; Hacioglu et al., 2003), ASSERT[5]. 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 et al., 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 which would be replaced by suitable question words. In case of a mandatory argument, the choice of question word depends on the argument's named entity tag (e.g. "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

3.3.1 Latent Dirichlet Allocation (LDA)

To measure the importance of the generated questions, we use LDA (Blei et al., 2003) to identify the important sub-topics 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.

[5]Available at http://cemantix.org/assert.html

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

Table 2: Semantic roles with possible question words

Here each topic is a probability distribution over words. LDA assumes that documents are made up of words and word ordering is not important ("bag-of-words" assumption) (Misra et al., 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 i^{th} 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 et al., 2003). A Dirichlet (α) prior is placed on θ and a Dirichlet (β) prior is set on ϕ to refine this basic model (Blei et al., 2003; Griffiths and Steyvers, 2002). 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[6] that uses the popular MALLET (McCallum, 2002) toolkit for the back-end. 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 sub-topics. For example, from the associated body of texts of the topic *Apple Inc. Logos*, we get these sub-topics: *janoff, themes, logo, color, apple*.

3.3.2 Extended String Subsequence Kernel (ESSK)

Once we identify the sub-topics, we apply ESSK to measure their similarity with the generated questions. ESSK is the simple extension of the Word Sequence Kernel (WSK) (Cancedda et al., 2003) and String Subsequence Kernel (SSK) (Lodhi et al., 2002). WSK receives two sequences of words as input and maps each of them into a high-dimensional vector space. WSK's value is just the inner product of the two vectors. But, WSK disregards synonyms, hyponyms, and hypernyms. On the other hand, SSK measures the similarity between two sequences of

[6]Available at http://code.google.com/p/topic-modeling-tool/

"alphabets". In ESSK, each "alphabet" in SSK is replaced by a disjunction of an "alphabet" and its alternative (Hirao et al., 2003). In ESSK, each word in a sentence is considered an "alphabet", and the alternative is its all possible senses. However, our ESSK implementation considers the alternative of each word as its disambiguated sense. We use a dictionary based Word Sense Disambiguation (WSD) System 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 and used 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, since WordNet orders the senses of a word by decreasing order of their frequency.

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 $\mathrm{Sim}(T_i, Q_j)$ 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[7]:

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

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

Here, $K'_m(t_i, q_j)$ is defined below. t_i and q_j are nodes of T and Q, respectively. The function $val(t, q)$ returns the number of attributes common (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:

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

Finally, the similarity measure is defined after normalization as below:

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

[7]The formulae denotes 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)

3.4 Judging Syntactic Correctness

The generated questions might be syntactically incorrect due to the process of automatic question generation. It is time consuming and a lot of 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. Hence, to judge the syntactic correctness of each generated question automatically, we apply the tree kernel functions and re-implement the syntactic tree kernel model 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[8] (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.

Each tree T is represented by an m dimensional vector $v(T) = \big(v_1(T), v_2(T), \cdots v_m(T)\big)$, where the i-th element $v_i(T)$ is the number of occurrences of the i-th tree fragment in tree T. The tree fragments of a tree are all of its sub-trees which include at least one production with the restriction that no production rules can be broken into incomplete parts. Figure 1 shows an example tree and a portion of its subtrees.

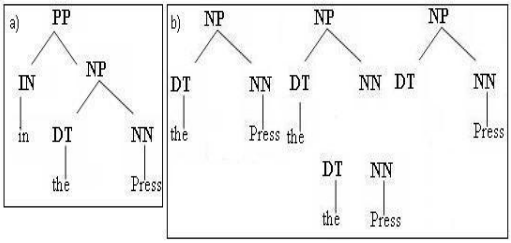

Figure 1: (a) An example tree (b) The sub-trees of the NP covering "the press".

Implicitly we enumerate all the possible tree fragments $1, 2, \cdots, m$. These fragments are the axis of this m-dimensional space. Note that this could be done only implicitly, since the number m is extremely large. Because of this, Collins and Duffy Collins and Duffy (2001) defined the tree kernel algorithm whose computational complexity does not depend on m. The tree kernel of two syntactic trees T_1 and T_2 is actually the inner product of $v(T_1)$ and $v(T_2)$:

$$TK(T_1, T_2) = v(T_1).v(T_2) \tag{2}$$

We define the indicator function $I_i(n)$ to be 1 if the sub-tree i is seen rooted at node n and 0 otherwise. It follows:

$$v_i(T_1) = \sum_{n_1 \in N_1} I_i(n_1)$$

[8] Available at ftp://ftp.cs.brown.edu/pub/nlparser/

$$v_i(T_2) = \sum_{n_2 \in N_2} I_i(n_2)$$

where, N_1 and N_2 are the set of nodes in T_1 and T_2 respectively. So, we can derive:

$$\begin{aligned} TK(T_1, T_2) &= v(T_1).v(T_2) \\ &= \sum_i v_i(T_1) v_i(T_2) \\ &= \sum_{n_1 \in N_1} \sum_{n_2 \in N_2} \sum_i I_i(n_1) I_i(n_2) \\ &= \sum_{n_1 \in N_1} \sum_{n_2 \in N_2} C(n_1, n_2) \end{aligned} \tag{3}$$

where, we define $C(n_1, n_2) = \sum_i I_i(n_1) I_i(n_2)$. Next, we note that $C(n_1, n_2)$ can be computed in polynomial time, due to the following recursive definition:

1. If the productions at n_1 and n_2 are different then $C(n_1, n_2) = 0$
2. If the productions at n_1 and n_2 are the same, and n_1 and n_2 are pre-terminals, then $C(n_1, n_2) = 1$
3. Else if the productions at n_1 and n_2 are not pre-terminals,

$$C(n_1, n_2) = \prod_{j=1}^{nc(n_1)} (1 + C(ch(n_1, j), ch(n_2, j))) \tag{4}$$

where, $nc(n_1)$ is the number of children of n_1 in the tree; because the productions at n_1 and n_2 are the same, we have $nc(n_1) = nc(n_2)$. The i-th child-node of n_1 is $ch(n_1, i)$.

Note that, the tree kernel (TK) function computes the number of common subtrees between two trees. Such subtrees are subject to the constraint that their nodes are taken with all or none of the children they have in the original tree. The *TK* (tree kernel) function gives the similarity score between each sentence in the given body of texts and the generated question based on the syntactic structure. Each sentence[9] 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 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 formula as follows:

$$w * ESSK_{score} + (1 - w) * SYN_{score} \tag{5}$$

Here w is the importance parameter which holds the value in $[0, 1]$. We kept $w = 0.5$ to give equal importance[10] to topic relevance and syntactic correctness.

[9] We consider that a question is syntactically fluent as well as relevant to the topic if it has similar syntactic sub-trees as those of the most sentences in the body of texts.

[10] A syntactically incorrect question is not useful even if it is relevant to the topic. This motivated us to give equal

4.2 Corpus

To run our experiments, we use the dataset provided in the Question Generation Shared Task and Evaluation Challenge[11] (QGSTEC, 2010) for the task of question generation from paragraphs. This dataset 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 dataset 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 use 10 topics and their associated paragraphs as the development data[12]. A total of 2186 questions are generated from the remaining 50 topics (test data) to be ranked.

4.3 Evaluation Setup

4.3.1 Methodology

We use a methodology derived from Boyer and Piwek (2010); Heilman and Smith (2010b) to evaluate the performance of our QG systems. Three native English-speaking university graduate students judge[13] 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. For each question, we calculate the average of the judges' scores.

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 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 sub-topics from the sentences present in the body of texts. The sub-topics are the important words in the context which are closely related to the topic and have significantly greater probability of occurring in the given text compared to that in a large background corpus. We use a topic signature computation tool[14] for this purpose. The background corpus that is used in this tool contains 5000 documents from the English GigaWord Corpus. For example, from the given body of texts of the topic *Apple Inc.*

importance to topic relevance and syntactic correctness. The parameter w can be tuned to investigate its impact on the system performance.

[11] http://www.questiongeneration.org/mediawiki

[12] We use this data to build necessary general purpose rules for our QG model.

[13] The inter-annotator agreement of Fleiss' $\kappa = 0.41, 0.45, 0.62$, and 0.33 are computed for the three judges for the results in Table 3 to Table 6, indicating moderate (for the first two tables), substantial and fair agreement (Landis and Koch, 1977) between the raters, respectively.

[14] Available at http://www.cis.upenn.edu/ lannie/topicS.html

Logos, we get these sub-topics: *jobs, logo, themes, rainbow, monochromatic*. Then we use the same steps of Section 3.3.2 and Section 3.4, and use equation 5 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[15] 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[16] 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 61.86%, and 34.98%, respectively, and improves the topic relevance and syntactic correctness scores over the *Baseline2* system by 7.40%, and 7.57%, 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 3.88%, and 2.89%, respectively. From these results, we can clearly observe the effectiveness of our proposed QG system. The improvements in the results are statistically significant[17] ($p < 0.05$).

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 were also useful to provide variety in the question space. We generated these questions using the NE information. As the performance of the NE-taggers were not that great, we had a few of these questions generated. In most cases, these questions were outranked by other important questions that included a combination of topics and sub-topics 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.

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

Table 3: Topic relevance and syntactic correctness scores

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

[15] Available at http://www.ark.cs.cmu.edu/mheilman/questions/

[16] We ignore the yes-no questions for our task.

[17] We tested statistical significance using Student's t-test.

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 more number of questions which proves the effectiveness of our QG system.

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 4: Acceptability of the questions (in %)

Systems	Topic Relevance	Syntactic Correctness
Baseline1 (No Ranking)	3.20	3.54
Baseline2 (Topic Signature)	3.80	3.92
State-of-the-art (Heilman and Smith, 2010b)	4.01	4.15
Proposed QG System	4.12	4.25

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

Systems	Top 15%	Top 30%
Baseline1 (No Ranking)	41.3	37.1
Baseline2 (Topic Signature)	53.5	43.6
State-of-the-art (Heilman and Smith, 2010b)	57.5	43.2
Proposed QG System	58.4	44.5

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

Narrowing Down the Focus We run further experiments by narrowing down the topic focus. We consider only the topics about persons (biographical focus). We choose 10 persons 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[18]. We generate a total of 390 questions from the considered 10 topics 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. Again, 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 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. 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.

[18] http://en.wikipedia.org/wiki/Time_100

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

Table 7: System output

4.3.4 An Input-Output Example

An input to our systems is for instance, 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.

Conclusion and Future Work

In this paper, we have considered the task of automatically generating questions from topics where each topic is associated with a body of texts containing useful information. We have exploited the named entity and semantic role labeling information to accomplish the task. A key aspect of our approach is the use of LDA to automatically discover the hidden sub-topics from the sentences. We have proposed a method to rank the generated questions by considering: 1) sub-topical 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 QG system with two baseline systems and one state-of-the-art system. The evaluation results have shown that the proposed QG system significantly outperforms all other considered systems as our system generated top-ranked questions are found to be better in topic-relevance and syntactic correctness than those of the other systems. We have conducted another experiment by narrowing down the topic focus. In this experiment, we have considered *persons* as topics. Our experiments have demonstrated the effectiveness of the proposed topic to question generation approach. We hope to carry on this ideas and develop further mechanisms to question generation based on the dependency features of the answers and answer finding (Li and Roth, 2006; Pinchak and Lin, 2006).

Acknowledgments

The research reported in this paper was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada – discovery grant and the University of Lethbridge.

References

Agarwal, M., Shah, R., and Mannem, P. (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. ACL.

Ali, H., Chali, Y., and Hasan, S. A. (2010). Automation of Question Generation from Sentences. In *Proceedings of QG2010: The Third Workshop on Question Generation*, Pittsburgh, USA.

Andrenucci, A. and Sneiders, E. (2005). Automated Question Answering: Review of the Main Approaches. In *Proceedings of the 3rd International Conference on Information Technology and Applications (ICITA'05)*, Sydney, Australia.

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

Boyer, K. E. and Piwek, P. (2010). Proceedings of QG2010: The Third Workshop on Question Generation. Pittsburgh: questiongeneration.org.

Brown, J. C., Frishkoff, G. A., and Eskenazi, M. (2005). Automatic Question Generation for Vocabulary Assessment. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Vancouver, British Columbia, Canada.

Cancedda, N., Gaussier, E., Goutte, C., and Renders, J. M. (2003). Word Sequence Kernels. *Journal of Machine Learning Research*, 3:1059–1082.

Celikyilmaz, A., Hakkani-Tur, D., and Tur, G. (2010). LDA based Similarity Modeling for Question Answering. In *Proceedings of the NAACL HLT 2010 Workshop on Semantic Search*, SS '10, pages 1–9. ACL.

Chali, Y., Hasan, S. A., and Joty, S. R. (2009a). 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, Singapore.

Chali, Y., Hasan, S. A., and Joty, S. R. (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., Joty, S. R., and Hasan, S. A. (2009b). Complex Question Answering: Unsupervised Learning Approaches and Experiments. *Journal of Artificial Intelligence Research*, 35:1–47.

Charniak, E. (1999). A Maximum-Entropy-Inspired Parser. In *Technical Report CS-99-12*, Brown University, Computer Science Department.

Chen, W., Aist, G., and Mostow, J. (2009). Generating Questions Automatically from Informational Text. In *Proceedings of the 2nd Workshop on Question Generation (AIED 2009)*, pages 17–24.

Collins, M. and Duffy, N. (2001). Convolution Kernels for Natural Language. In *Proceedings of Neural Information Processing Systems*, pages 625–632, Vancouver, Canada.

Echihabi, A. and Marcu, D. (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. ACL.

Fellbaum, C. (1998). WordNet - An Electronic Lexical Database. Cambridge, MA. MIT Press.

Geman, S. and Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6:721–741.

Graesser, A. C., VanLehn, K., Rose, C. P., Jordan, P. W., and Harter, D. (2001). Intelligent Tutoring Systems with Conversational Dialogue. *AI Magazine*, 22(4):39–52.

Griffiths, T. L. and Steyvers, M. (2002). Prediction and Semantic Association. In *NIPS'02*, pages 11–18.

Hacioglu, K., Pradhan, S., Ward, W., Martin, J. H., and Jurafsky, D. (2003). Shallow Semantic Parsing Using Support Vector Machines. In *Technical Report TR-CSLR-2003-03*, University of Colorado.

Heilman, M. and Smith, N. A. (2010a). Extracting Simplified Statements for Factual Question Generation. In *Proceedings of the Third Workshop on Question Generation*.

Heilman, M. and Smith, N. A. (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.

Hickl, A., Lehmann, J., Williams, J., and Harabagiu, A. (2005). Experiments with Interactive Question-Answering. In *In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 60–69.

Hirao, T., Suzuki, J., Isozaki, H., and Maeda, E. (2003). NTT's Multiple Document Summarization System for DUC2003. In *Proceedings of the Document Understanding Conference*.

Hirao, T., Suzuki, J., Isozaki, H., and Maeda, E. (2004). Dependency-based Sentence Alignment for Multiple Document Summarization. In *Proceedings of COLING 2004*, pages 446–452, Geneva, Switzerland. COLING.

Ignatova, K., Bernhard, D., and Gurevych, I. (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. NSF.

Kingsbury, P. and Palmer, M. (2002). From Treebank to PropBank. In *Proceedings of the International Conference on Language Resources and Evaluation*, Las Palmas, Spain.

Kotov, A. and Zhai, C. (2010). Towards Natural Question Guided Search. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 541–550. ACM.

Landis, J. R. and Koch, G. G. (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159–174.

Lauer, T. W., Peacock, E., and Graesser, A. C. (1992). Questions and Information Systems.

Li, X. and Roth, D. (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. NSF.

Lin, C. Y. and Hovy, E. H. (2000). The Automated Acquisition of Topic Signatures for Text Summarization. In *Proceedings of the 18th conference on Computational linguistics*, pages 495–501.

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., and Watkins, C. (2002). Text Classification using String Kernels. *Journal of Machine Learning Research*, 2:419–444.

Mannem, P., Prasad, R., and Joshi., A. (2010). Question Generation from Paragraphs at Upenn. In *Proceedings of the Third Workshop on Question Generation*.

McCallum, A. K. (2002). MALLET: A Machine Learning for Language Toolkit.

McConnell, C. C., Mannem, P., Prasad, R., and Joshi, A. (2011). A New Approach to Ranking Over-Generated Questions. In *Proceedings of the AAAI Fall Symposium on Question Generation*.

McGough, J., Mortensen, J., Johnson, J., and Fadali, S. (2001). A Web-based Testing System with Dynamic Question Generation. In *ASEE/IEEE Frontiers in Education Conference*.

Misra, H., Cappé, O., and Yvon, F. (2008). Using LDA to Detect Semantically Incoherent Documents. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, CoNLL '08, pages 41–48. ACL.

Mitkov, R. and Ha, L. A. (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.

Moschitti, A. and Basili, R. (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*, Genoa, Italy.

Moschitti, A., Quarteroni, S., Basili, R., and Manandhar, S. (2007). Exploiting Syntactic and Shallow Semantic Kernels for Question/Answer Classificaion. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 776–783, Prague, Czech Republic. ACL.

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

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

Pasca, M. and Harabagiu, S. M. (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, France.

Pinchak, C. and Lin, D. (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.

Ratinov, L. and Roth, D. (2009). Design Challenges and Misconceptions in Named Entity Recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. ACL.

Rus, V., Cai, Z., and Graesser, A. C. (2007). Experiments on Generating Questions About Facts. In *Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Text Processing*, pages 444–455. Springer-Verlag.

Rus, V. and Graesser, A. C. (2009). The Question Generation Shared Task and Evaluation Challenge. In *Workshop on the Question Generation Shared Task and Evaluation Challenge, Final Report*, The University of Memphis. National Science Foundation.

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. NSF.

Wang, W., Tianyong, H., and Wenyin, L. (2008). Automatic Question Generation for Learning Evaluation in Medicine. In *LNCS Volume 4823*.

Wei, X. and Croft, W. B. (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. ACM.

Zhang, A. and Lee, W. (2003). Question Classification using Support Vector Machines. In *Proceedings of the Special Interest Group on Information Retrieval*, pages 26–32, Toronto, Canada. ACM.

Zheng, Z., Si, X., Chang, E. Y., and Zhu, X. (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.