Cloze and Open Cloze Question Generation Systems and their Evaluation Guidelines

Thesis submitted in partial fulfillment
of the requirements for the degree of

MS by Research
in
Computer Science with specialization in NLP

by

Manish Agarwal
200702020
manish.agarwal@research.iiit.ac.in

Language Technology and Research Center (LTRC)
International Institute of Information Technology
Hyderabad - 500032, INDIA
July 2012

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

This is to certify that the thesis entitled **“Cloze and Open Cloze Question Generation Systems and their Evaluation Guidelines”’** submitted by **Manish Agarwal** to the International Institute of Information Technology, Hyderabad, for the award of the Degree of **Master of Science (by Research)** is a record of bona-fide research work carried out by him under my supervision and guidance. The contents of this thesis have not been submitted to any other university or institute for the award of any degree or diploma.

_____ _____

Date Adviser: Prof. Rajeev Sangal

To my Parents

Acknowledgments

First and foremost, I would like to thank Prof Rajeev Sangal who has had a tremendous influence on me. Apart from being my thesis advisor, he has been a dear friend, a profound philosopher, far beyond ordinary. This, together with his sharpness, energy and amazing sense of humour makes him a remarkable person and pleasure to work with. The questions he kept bombarding me with and the encouragement he provided helped me keep going. He always encouraged me to develop a broader perspective towards research and life in general. Today, as I look back, I find these discussions very close to my heart and by far the most cherished part of my journey. These interactions introduced me to a newer perspective of looking at things around me and have greatly influenced my understanding of life in general. He introduced me to the area of Natural Language Processing and closely guided my first steps. Some of the course he taught us - Natural Language Processing, Artificial Intelligence and the opportunity of being a Teaching Assistant to him for NLP played a major role in paving a way to this thesis.

Another person with whom I have been fortunate to be associated with is Mr. Prashanth Mannem. I would like to thank him for giving me the opportunity to be a Teaching Assistant for the course on Natural Language Processing, time he took out to expose me to the concepts in Natural Language Processing and Computational Linguistic, and being a constant support during the ups and the downs.

I am also thankful to all the faculty members of LTRC - Dr. Dipti Misra Sharma, Dr. V. Sriram and Dr. Soma Poul for providing a wonderful research center. I also take this opportunity to thank IIIT-Hyderabad for giving me an opportunity to see the world of research so closely. Without these people this thesis would not have been in such a good shape. I would also like to thank my seniors Avinesh PVS, Bharat Ram Ambati, Phani Gadde, GSK Chaitanaya, Meher Vijay for their guidance.

This acknowledgement would be incomplete without mention of my friends - Rakshit Shah, Rahul Agarwal, Ashish Jain, Karan Jindal, Rahul Goutam, Shashikant Muktiyar, Prudhvi Kosaraju, Shruthilaya, Abhinav Mehta, Animesh Chatterji, Abhijeet Gupta, Abinash Mahapatro, Manish Kumar Sharma, Ojasvi Rajpal, Pankaj Anthwal, Paridhi Rawat, Akshat Bakliwal, Piyush Arora, Ankit Patil, Rahul Kumar Namdev. I would like to thank them for their support, encouragement and foremost for making this journey memorable.

I would also like to acknowledge the undergraduates of the batch of 2009, to whom I was a Teaching Assistant for various courses. With them I took a deeper plunge in some of the topics in theoretical computer science.

Above all I would like to express my gratitude towards my parents who have had, are having and will continue to have a tremendous influence on my development. I am deeply thankful to my brother and sister for their eternal love and support.

There is a long list of people whom I fail to mention here, but have played their parts in bringing me to a level where I am capable of appreciating numerous wonders of computer science. I hold a deep gratitude towards all those.

The results presented in this thesis have been obtained in collaboration with Rakshit Shah.

Abstract

Question Generation is the task of generating questions from various inputs such as raw text or semantic representations. It is an attempt towards structuring the potential educational text which is available everywhere on the Internet. Generating questions manually on these large amounts of varied text is a tedious task. Hence, there is a need for generating questions automatically from these texts. The present work generates questions automatically from a given document. The aim is to build a system which can be used for different domains in various languages.

Questions can be broadly classified into Subjective Questions and Objective Questions. The work mainly focuses on generating two types of objective questions, (i) Cloze Questions and (ii) Open-cloze Questions. To present the method, English language texts have been used.

A cloze question contains a sentence with one or more blanks in it and four alternatives are given to fill those blanks. One of the four alternatives is correct and the others are wrong. The wrong alternatives are called distractors since they distract a student from choosing the correct answer. Cloze question generation is a three stage process, (i) Sentence Selection, (ii) Keyword Selection and (iii) Distractor Selection. In the presented system, all the three stages use heuristically weighted features to select relevant sentences, keywords and distractors.

Finding three relevant distractors corresponding to each selected keyword is the most difficult task. Therefore in the next attempt, a rule based open-cloze question generation system has been developed. Open-cloze questions are similar to cloze-questions without alternatives, which make them more difficult to solve. As it is difficult to evaluate the quality of questions generated automatically, this work also provides guidelines for the manual evaluation of automatically generated cloze and open cloze questions.

The present work generates questions that assess the content knowledge that a student has acquired upon reading a text rather than vocabulary, grammar assessment or language learning. There are only a few previous works under the area of QG, and this is the first attempt ever to generate cloze and open-cloze questions on content knowledge. Hence, a comparison between this system and the previous systems using a common dataset will not fetch any results. However, different individual stages of this method have been compared with previous approaches. In the comparison, it has been shown that the presented method covers more types of sentences to generate good quality questions. In addition, the method does not use any external resources unlike previous works and that makes it both domain and language independent.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A driving motivation for this work is to create tools to help teachers and students generate instructional content for different domains in various languages. We aim to create a system for Question Generation that can take as input a text article and create as output a list of questions on content knowledge.

1.1 Examples of Question Generation

In this section, we provide examples that illustrate that QG about explicit factual information is a challenging but still feasible task given current natural language processing (NLP) technologies. To explain question generation from a document, we begin with a relatively straight forward example, taken from a Wikipedia article about *Sachin Tendulkar*.

... Sachin Ramesh Tendulkar (born 24 April 1973) is an Indian cricketer widely regarded as one of the greatest batsmen in the history of cricket. He is the leading run-scorer and century maker in Test and one-day international cricket. He is the first male player to score a double century in ODI cricket. In 2002, just 12 years into his career, Wisden ranked him the second greatest Test batsman of all time, behind Donald Bradman, and the second greatest one-day-international (ODI) batsman of all time, behind Viv Richards. Tendulkar was an integral part of the 2011 Cricket World Cup winning Indian team at the later part of his career, his first such win in six World Cup appearances for India. Tendulkar is the first and the only cricketer to accomplish the feat of scoring a hundred centuries in international cricket which includes 49 ODI & 51 Test cricket centuries. ....

A number of different type of important questions can be generated based on above text, few example questions are presented below.

1. When was Sachin Tendulakar born?
2. In which year India won the world cup?

3. Who are the greatest batsman of all time in test and one-day cricket?
4. In how many world cups Sachin did play?
5. Give a brief idea about Sachin Tendulkar.
6. Sachin is the _____ player to score a double century in ODI.
   (a) First (b) Second (c) Third (d) Forth
7. How many centuries did Sachin score in ODIs.
   (a) 48 (b) 49 (c) 50 (d) 51

All the above questions are factual and came out of large discourse. Questions can only be answered after reading the above text. Many question-types like *when*, *who*, *which*, *what* and *how* are involved here. Also a *Cloze Question* (CQ, Question *6*) and a Multiple Choice Question (MCQ, Question *7*), where a important sentence is selected from the text then a keyword is blanked out and four alternatives are given (1 correct and 3 incorrect) to fill the blank.

1.2 Type of Questions

There can be many ways to ask a factual question. It could be an objective or a subjective question depends on its answer scope. So here we divided questions in two broad types, (i) Based on answer scope and (ii) Based on their presentation.

1.2.1 Based on Presentation

There are various ways possible to present a question. So in this class questions are classified based on their presentation style.

1.2.1.1 Fill-in-the-blank Question

Commonly known as objective questions. Here also two type of questions are possible (i) with alternatives, (ii) without alternatives. There are two methods to ask a objective question. If a sentence with one or more blanks (*question sentence*) are given with four alternatives to fill them, then the question is called *cloze-question* and without alternatives this is called *open-cloze* question. Question *6* is a example of a *cloze question*.

1.2.1.2 *wh-type* Question

Question begins with *wh-word*. *wh-words* are also called *question-words* (*q-words*), like *why, who, when, how,* etc. All above questions (*1,2,3,4* and *5*) can be categorized as a wh-type questions.

If instead of *question sentence* (sentence with one or more blanks) a proper well framed question with a *wh-word* is presented then that question will be called Multiple Choice Question (MCQ). MCQ Example Question *7*.

Manual construction of MCQs, however, is a time-consuming and labour-intensive task. As opposed to MCQs where one has to generate the WH style question, CQs use a sentence with blanks to form a question. The sentence could be picked from a document on the topic avoiding the need to generate a WH style question. No requirement of Natural Language Generation (NLG) and easy evaluation of these kind of questions are great benefits of this area. As a result, automatic CQG has received a lot of research attention recently.

1.2.2 Based on Answer Scope

In this category questions are categorized based on their answer scope in the article. Basically length of the answer span decides, in which category a question belongs.

1.2.2.1 Specific Scope

Other than above two works (for medium scope [3], for global scope [32]) all the previous works are tried to generate specific scope questions in various different types. Whether it is subjective wh-type questions, MCQs, CQs or *Open-cloze Questions* (OCQs).
Example: Question *1*,*2*,*4*,*6* and *7*. All these questions' answer are present with in a single clause.

1.2.2.2 Medium Scope

Question whose answer covers more than 1 clause or till 2, 3 clauses. These questions are also equally difficult as global scope questions because of the same reason. Example of these kind of question could be Question *3*. In Question *3*, scope the question's answer is *In 2002, just 12 years into his career, Wisden ranked him the second greatest Test batsman of all time, behind Donald Bradman, and the second greatest one-day-international (ODI) batsman of all time, behind Viv Richards*.

One of our work [3] has generated medium scope questions using discourse cues, described in Chapter-5.

1.2.2.3 Global Scope

A question is called a *global scope* question, when its answer's scope is a whole article. Because to answer the questions one must have read the article. Few standard global questions could be like Question *5*. Other than these specific format questions, asking a specifically global scope question would be very difficult. Because asking a single question on the article needs a full understanding of it, which seems impossible task today in NLP. Although there is one work [32] which generates global questions, will talk about the work in the next chapters.

1.3 QG from a Document

In this work we present a system which generates questions based on given document. There is huge difference between generating questions from a given sentence and a given document. When we generate questions from a sentence we just have to select a keyword and present a question on the keyword with few transformations. But when we generate questions from a document there are lot of things one has to consider. Now since we have full context we can't choose any keyword from any sentence in order to generate a question. Generated questions must be of good quality from the document. So one has to understand the document and based on that find out which keyword from which sentence can give a good question. So the task of question generation from the document involves, sentence selection and keywords selection. In sentence selection, relevant and informative sentences from the document should be selected. In keyword selection one must select questionable keywords from those selected sentences. Not only a keyword should be important enough to ask a question upon, but also the generated question must be answerable.

Question generation from a document also implies generation of factual questions. Questions can be generated to test various knowledge like, grammar, preposition, vocabulary, etc. So when we say, generating questions from a document it means questions based on document knowledge. These kind of questions called factual questions. The document could be a chapter of a book, of any domain or a news article from today's newspaper, any English article on which you want to test someone's knowledge. Since the task of generating questions manually is time taking and boring for a human being, automation is required. A automatic method can generate these questions within fraction of seconds anytime. So one can judge himself before the examination or a teacher could generate the questions for a class exam quickly.

1.4 Educational Value of QG

There are various application of the task, like generating lot of frequently asked questions automatically, online quizzes, examination papers, etc. A QG system would be useful for building an automated trainer for learners to ask better questions, and for building better hint and 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. Available studies revealed that humans are not very skilled in asking good questions. Many intelligent tutoring systems can be seen as a tool for teachers in that they provide guidance and feedback while students work through practice exercises. In facilitating practice, tutoring systems allow teachers to focus their effort on other issues such as planning curricula and delivering instruction about new concepts.

By focusing on explicit factual information, we are restricting the range of useful questions we might be able to produce. Factual questions make up only a small part of the questions used by educators in practice exercises and assessments. For example, educators also ask questions to assess the ability

of students to make inferences, as well as their ability to perform various reading strategies such as summarizing or making connections to prior knowledge. However, automatic factual QG may still have the potential to help teachers create instructional content. Manual generation of questions for these kind of application is time taking and sometime boring too. If we have an automatic question generation system which can do this task in few seconds will help us a lot. Other than educational applications, in trivia games, assigning fan ratings on social networks by posing game related questions etc automatic question generation system is very useful.

1.5 Primary Contribution and Thesis Statement

This section summarizes the main contributions of this research and presents a concise thesis statement.

1.5.1 Summary of Primary contribution

1. Cloze question generation system based on heuristic weights without using any external resources.
2. Open-cloze question generation system using rule based techniques.
3. Evaluation guidelines for manual evaluation of the Cloze and Open Cloze questions.

1.5.2 Thesis Statement

Build an automatic question generation system which can generate different kind of important factual questions based on any document.

1.6 Chapterization

In chapter-2, we list the previous works in this area and also compare our work with them. In chapter-3, we present our factual cloze question generation tool, which generates questions from a biology text book through heuristically weighted features. We do not use any external knowledge and rely only on information present in the document to generate the CQs with distractors. No use of external resources makes our system unique. In chapter-4, we present our OQG (open-cloze question generation) system, with evaluation guidelines for these type of questions. Since there is neither automatic evaluation tool for questions nor specific guidelines to evaluate them manually. In chapter-5, we discuss about the medium scope questions. We will present challenges and then describe approach to generate this kind of questions. Finally we conclude the things in the last chapter of the thesis.

Chapter 2

Related Work on Question Generation

Ideal learners are often curious question generators who actively self-regulate their learning. That is, they identify their own knowledge deficits, ask questions that focus on these deficits, and answer the questions by exploring reliable information sources. Unfortunately, this idealistic vision of intelligent inquiry is rarely met, as most learners have trouble identifying their own knowledge deficits [53]. Automatic Question Generation from sentences and paragraphs has caught the attention of the NLP community in the last few years through the question generation workshops and the shared task in 2010 [2]. Manual generation of questions takes lot of time and human efforts too, which is ineffective and attempts have been made to do this task online.

Question asking and Question Generation (QG) are important components in advanced learning technologies such as intelligent tutoring systems, and inquiry-based environments. QG is an essential element of learning environments, help systems, information seeking systems, and a myriad of other applications [25].

In [41] they introduced a template based approach to generate questions on four types of entities. The authors in [20] used WTML (Web Testing Markup Language), which is an extension of HTML (Hyper Text Markup Language), to solve the problem of presenting students with dynamically generated browser-based exams with significant engineering mathematics content. In [65], they generated the questions automatically based on question templates that are created by training on many medical documents. In [10], an interesting approach was described to automatically generating questions for vocabulary assessment. Many applications of computer technology for assisting teachers involve hardware, such as clicker devices for gathering student responses [62] and smartboards [55], or standard productivity software such as Microsoft PowerPoint [60].

The study of artificial intelligence in educational technologies, particularly of intelligent tutoring systems [23], [63], [23], [51] and [66] is a growing field. Many intelligent tutoring systems can be seen as a tool for teachers in that they provide guidance and feedback while students work through practice exercises. In facilitating practice, tutoring systems allow teachers to focus their effort on other issues such as planning curricula and delivering instruction about new concepts. It is worth noting that much of the work on tutoring systems has focused on interactions between an individual student and the

computer, and less research on how tutoring systems affect teachers though there is definitely research on such issues, particularly work by [13] and [5]. For example, [52] describes an authoring tool for automatically parsing the text of an algebra word problem into a formal semantic representation that could be loaded into a cognitive tutor.

Some of the challenges of authoring content can be addressed by intelligently re-using work by human teachers. For example, [6] describe a crowd sourcing approach to the problem of generating content. Focusing on math exercises, they propose creating an online repository of materials by eliciting contributions from teachers and other Internet users. Such content could potentially be used within a tutoring system, or perhaps more directly by classroom teachers. Note that in such an online system for sharing or creating content, automated techniques such as the QG system we describe here could provide seed content for contributors to select and revise.

There are also works on automatically creating content in the area of computer-assisted language learning. For example, [34] describe a system that takes arbitrary texts as input and, with NLP technologies, highlights specific grammatical constructions and automatically creates grammar practice exercises. Also, [17] describe a system that uses natural language processing and text retrieval technologies to help English as a Second Language teachers find pedagogically appropriate reading practice materials (e.g., texts at an appropriate reading level) for intermediate and advanced language learners. There has been considerable work on applying NLP to educational problems, but most applications deal with the analysis of student responses rather than the generation of instructional content.

For example, tutorial dialogue systems [31, 15, 8] use NLP to analyze students dialogue moves and respond in pedagogically appropriate ways. Automated scoring technologies [54, 39, 42] grade student responses to essays or short answer questions. Systems for grammatical error detection [26] analyze student writing to find errors involving prepositions, determiners, etc. And, finally, tools for analyzing discussion boards [33] use NLP to provide teachers with efficient ways of monitoring discussions among large groups of students.

Most of the works in this area has concentrated on generating questions from individual sentences [64, 44, 19]. [57] used question templates and [17] used general-purpose rules to transform sentences into questions. A notable exception is [32] who generated questions of various scopes (general, medium and specific). They used a fixed template to generate a global answer scope questions from first sentence of wikipedia documents. There is one our work for generating medium scope questions, [4]. We have used discourse connectives to generate this type of questions, explained in Chapter-5.

Other than subjective questions there are works for generating objective question (Cloze and Open-cloze question). The simplest method to generate a cloze question, one takes a passage and replaces some of the words in the passage with blanks, for example choose every Nth word. The readers task is then to identify the original words that fill in the blanks. Researchers have found that such cloze questions are effective for measuring first language reading comprehension ([7, 50]) as well as second language ability ([43]). Cloze tests can also be automated without introducing many errors (see, e.g., [40]) since they only require tokenization of an input text into words.

Works in Cloze question generation [59, 28, 30, 45, 56], [21], [38] have mostly worked in the domain of English language learning. Cloze-question have been generated to test student's knowledge of English in using the correct verbs [59], prepositions [28] and adjectives [30] in sentences. [45] and [56] have generated CQs to teach and evaluate student's vocabulary.

In this thesis we have presented two works where we generate fill-in-the-blank questions. In the first work we have used heuristic features' to weight different features in different modules of the system. In the next work we tried to generate Open-Cloze questions using rule based methods in first two modules, *Sentence Selection* and *Keyword Selection*.

Chapter 3

Heuristic Based Models for Cloze Question Generation System

3.1 Introduction

In this chapter, a Cloze-question Generation System (CQG) is presented, which generates questions from a document. Cloze questions are fill-in-the-blank questions with multiple choices (one correct and three incorrect answers) provided. The system finds the informative and relevant sentences from the document and generates cloze questions from them by first blanking keywords from the sentences and then determining the distractors for these keywords. These questions, being multiple choice ones, are easy to evaluate and don't need any Natural Language Generation (NLG) technique in their generation process.

Syntactic and lexical features are used in this process without relying on any external resource apart from the information in the document. Evaluation is done on two chapters of a standard biology textbook and presented the results. Preparing these questions manually will take a lot of time and effort. This is where automatic *cloze question generation* (CQG) from a given document is useful.

3.1.1 Terminology

1. *A _____ bond is the sharing of a pair of valence electrons by two atoms.*
   (a) Hydrogen (b) Covalent (c) Ionic (d) Double (correct answer: Covalent)

In a Cloze Question (CQ) such as the one above Example 1, we refer to the sentence with the gap as the *question sentence* (QS) and the sentence in the text that is used to generate the QS as the cloze sentence (CS). The word(s) which is removed from a CS to form the QS is referred to as the keyword while the three alternatives in the question are called as *distractors*, as they are used to distract the students from the correct answer.

In this work, we move away from the domain of English language learning and work on generating cloze questions from the chapters of a biology textbook used for Advanced Placement (AP) exams.

The aim is to go through the textbook, identify *informative sentences*[1] and generate cloze questions from them to aid students' learning. The system scans through the text in the chapter and identifies the *informative sentences* in it using features inspired by summarization techniques. Questions from these sentences (CSs) are generated by first choosing a keyword in each of these and then finding appropriate distractors for them from the chapter.

A document with its title is taken as input and a list of cloze questions is presented as output. Unlike previous works [9, 56], no external resource are used for distractor selection, making it adaptable to text from any domain. Its simplicity makes it useful not only as an aid for teachers to prepare cloze questions but also for students who need an automatic question generator to aid their learning from a textbook.

3.2 Data Used

A Biology text book *Campbell Biology, 6th Edition* has been used for this work. Experiments are done using 2 chapters *(the structure and function of macromolecules* and *an introduction to metabolism)* of unit 1 from the book. Each chapter contains sections and subsections with their respective topic headings. Number of subsections, sentences, average words per sentence in each chapter are (25, 416, 18.3) and (32, 423, 19.5) respectively. Each subsection is taken as a document. The chapters are divided into documents and each document is used for CQG independently.

3.3 Approach

Given a document, the CQs are generated from it in three stages: *Sentence Selection, Keyword Selection* and *Distractor Selection*. *Sentence Selection* involves identifying informative sentences in the document which can be used to generate a CQ. These sentences are then processed in the *Keyword Selection* stage to identify the keyword to ask the question on. In the final stage, the distractors for the selected keyword are identified from the given chapter by searching for words with the same context as that of the keyword.

In each stage, the system identifies a set of candidates (i.e. all sentences in the document in stage I, words in the previously selected sentence in stage II and words in the chapter in stage III) and extracts a set of features relevant to the task. *Weighted sum of extracted features* (see equation 3.1) is used to score these candidates, with the weights for the features in each of the three steps assigned heuristically. A small development data has been used to tune the feature weights.

$$score = \sum_{i=0}^{n} w_i \times f_i \tag{3.1}$$

In equation 3.1, f_i denotes the feature and w_i denotes the weight of the feature f_i. The overall architecture of the system is shown in Figure 3.3.

[1] A sentence is deemed informative if it has the relevant course knowledge which can be questioned.

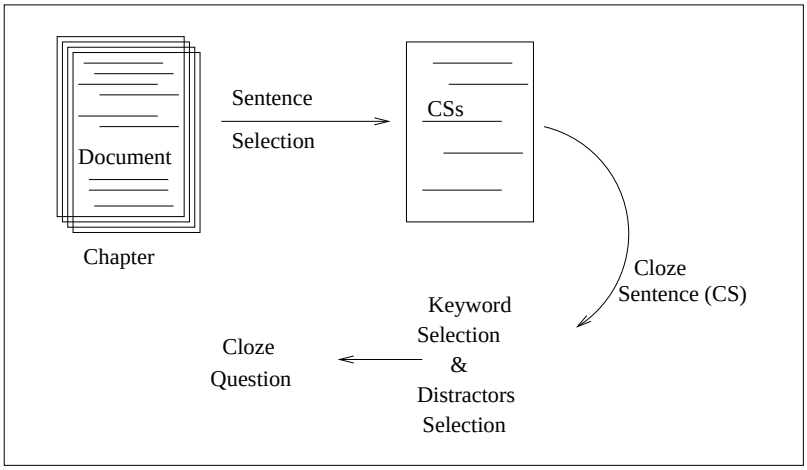

Figure 3.1 System Architecture

In earlier approaches to generating cloze questions (for English language learning), the keywords in a text were gathered first (or given as input in some cases) and all the sentences containing the keyword were used to generate the question. In domains where language learning is not the aim, a cloze question needs an informative sentence and not just any sentence with the desired keyword present in it. For this reason, in our work, *Sentence Selection* is performed before *Keyword Selection*.

3.3.1 Sentence Selection

A good CS should be (1) *informative* and (2) *cloze question-generatable*. An informative sentence in a document is one which has relevant knowledge that is useful in the context of the document. A sentence is *cloze question-generatable* if there is sufficient context within the sentence to predict the keyword when it is blanked out. An *informative sentence* might not have enough context to generate a question from and vice versa.

The *Sentence Selection* module goes through all the sentences in the documents and extracts a set of features from each of them. These features are defined in such a way that the two criterion defined above are accounted for. Table 3.1 gives a summary of the features used.

First sentence: $f(s_i)$ is a binary feature to check whether the sentence s_i is the first sentence of the document or not. Upon analyzing the documents in the textbook, it was observed that the first sentence in the document usually provides a summary of the document. Hence, $f(s_i)$ has been used to make use of the summarized first sentence of the document. So for the first sentence of a document feature value will be *1* and for other sentences' it will be *0*.

Feature Symbol	Description	Criterion
f(s_i)	Is s_i the first sentence of the document?	**I**
sim(s_i)	No. of tokens common in s_i and title / length(s_i)	**I, G**
abb(s_i)	Does s_i contain any abbreviation?	**I**
super(s_i)	Does s_i contain a word in its superlative degree?	**I**
pos(s_i)	s_i's position in the document (= i)	**G**
discon(s_i)	Is s_i beginning with a discourse connective?	**G**
l(s_i)	Number of words in s_i	**G**
nouns(s_i)	No. of nouns in s_i / length(s_i)	**G**
pronouns(s_i)	No. of pronouns in s_i / length(s_i)	**G**

Table 3.1 Feature set for *Sentence Selection* (s_i: i^{th} sentence of the document; **I**: to capture *informative sentences*; **G**: to capture the potential candidate for generating a CQs)

Common tokens: *sim*(s_i) is the count of words (nouns and adjectives) that the sentence and the title of the document have in common. A sentence with words from the title in it is important and is a good candidate to ask a question using the common words as the keyword.

2. *The different states of potential **energy** that **electrons** have in an atom are called **energy levels**, or **electron** shells.* (Title: *The Energy Levels of Electrons*)

In Example 2, value of the feature is 3/19 (common words:3, sentence length:19) and generating cloze question using *energy, levels* or *electrons* as the keyword will be useful.

Abbreviations and Superlatives: $abb(s_i)$, $super(s_i)$ features capture those sentences which contain abbreviations and words in superlative degree respectively. The binary features determine the degree of the importance of a sentence in terms of the presence of abbreviations and superlatives.

3. *In living organisms, most of the **strongest** chemical bonds are covalent ones.*

In Example 3, presence of *strongest* makes sentence more informative and relevant, therefore useful for generating a CQ.

Sentence position: $pos(s_i)$ is position of the sentence s_i, in the document (= i). Since topic of the document is elaborated in the middle of the document, the sentences occurring in the middle of the document are less important for the CQs than those which occur either at the start or the end of the document. In order to use the above observation, the module uses this feature.

Discourse connective at the beginning: $discon(s_i)$'s value is *1* if first word of s_i is a *discourse connective*[2] and *0* otherwise. Discourse connective at the beginning of a sentence indicates that the sentence might not have enough context for a QS to be understood by the students.

4. *Because of this, it is both an **amine** and a **carboxylic** acid.*

In Example 4, after selecting *amine* and *carboxylic* as a keyword, QS will be left with insufficient context to answer. Thus binary feature, $discon(s_i)$, is used.

Length: $l(s_i)$ is the number of words in the sentence. It is important to note that a very short sentence might generate an unanswerable question because of short context and a very long sentence might have enough context to make the question generated from it trivial.

Number of nouns and pronouns: Features $nouns(s_i)$ and $pronouns(s_i)$ represent the amount of context present in a sentence. More number of pronouns in a sentence reduces the contextual information, instead more number of nouns increases the number of potential keywords to ask a cloze question on.

Four sample CSs are shown in Table 3.2 with their document's titles.

No.	Selected Sentences
1	An electron having a certain discrete amount of energy is something like a ball on a staircase. *(The Energy Levels of Electrons)*
2	Lipids are the class of large biological molecules that does not include polymer. *(Lipids–Diverse Hydrophobic Molecules)*
3	A DNA molecule is very long and usually consists of hundreds or thousands of genes. *(Nucleic acids store and transmit hereditary information)*
4	The fatty acid will have a kink in its tail wherever a double bond occurs. *(Fats store large amounts of energy)*

Table 3.2 Selected Sentences from the different Documents

3.3.2 Keyword Selection

For each selected sentence in the previous stage, the *Keyword Selection* stage identifies the most appropriate keyword from the sentence to ask the question on. There are various ways of choosing words to replace, the simplest being to choose every N th word.

[2] *because, since, when, thus, however, although, for example* and *for instance* connectives have been included.

Previous works in this area, Smith et al. [56] take keywords as input and, Karamanis et al. [21] and Mitkov et al. [38] select keywords on the basis of term frequency and regular expressions on nouns. Then they search for sentences which contain that particular keyword in it. Since their approaches generate cloze questions only with one blank, they could end up with a trivial CQ, especially in case of conjunctions.

5. *Somewhere in the transition from molecules to cells, we will cross the blurry boundary between* ***nonlife and life****.*

For instance in Example 5, selecting only one of *non-life* and *life* makes the question trivial. This is an other reason for performing *Sentence Selection* before *Keyword Selection*. Our system can generate CQs with multiple blanks unlike previous works described above.

The *Keyword Selection* from a CS is two step process. In the first step the module generates a list of potential keywords from the CS (*keyword-list*) and in the second step it selects the best keyword from this *keyword-list*.

3.3.2.1 Keyword-list formation

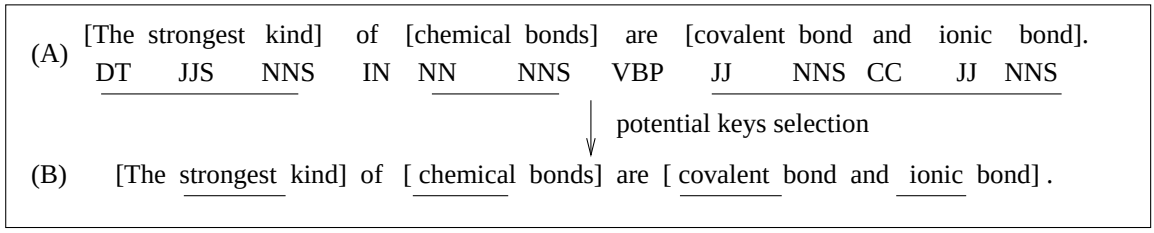

Figure 3.2 Generating *potential keyword*'s list, (*keyword-list*) of *strongest, chemical* and *covalent + ionic*.

A list of potential keywords is created in this step using the part of speech (POS) tags of words and chunks of the sentence in the following manner:

1. Each sequence of words in all the noun chunks is pushed into *keyword-list*. In Figure 3.3.2.1(A), the three noun chunks *the strongest kind*, *chemical bond* and *covalent bond and ionic bond* are pushed into the *keyword-list*.

2. For each sequence in the *keyword-list*, the most important word(s) is selected as the potential keyword and the other words are removed. The most important word in a noun chunk in the context of CQG in biology domain is a cardinal, adjective and noun in that order. In case where there are multiple nouns, the first noun is chosen as the potential keyword. If the noun chunk is a NP coordination, both the conjuncts are selected as a single potential keyword making it a case

of multiple gaps in QS. In Figure 3.3.2.1(B) potential *keywords strongest*, *chemical* and *covalent + ionic* are selected from the noun chunks by taking the order of importance into account.

An automatic POS tagger and a noun chunker has been used to process the sentences selected in the first stage. It was observed that if words of a keyword are spread across a chunk then there might not be enough context left in QS to answer the question. The noun chunk boundaries ensure that the sequence of words in the potential keywords are not disconnected.

6. *Hydrogen has 1 valence **electron** in the first shell, but the shell's capacity is 2 **electrons**.*

Any element of the *keyword-list* which occurs more than once in the CS is discarded as a potential keyword as it more often than not generates a trivial question. For instance, in Example 6 selecting any one of the two *electron* as a keyword generates an easy cloze question.

7. *In contrast , trypsin , a digestive enzyme residing in the alkaline environment of the intestine , has an optimal pH of _____.*
   (a) 6 (b) 7 (c) 8 (d) 9 (correct answer: 8)

If cardinals are present in a CS, the first one is chosen as its keyword directly and a cloze question has been generated (Example 7).

3.3.2.2 Best Keyword selection

In this step three features, $term(keyword_p)$, $title(keyword_p)$ and $height(keyword_p)$, described in Table 3.3, are used to select the best keyword from the *keyword-list*.

Feature Symbol	Description
$term(keyword_p)$	Number of occurrences of the $keyword_p$ in the document.
$title(keyword_p)$	Does title contain $keyword_p$?
$height(keyword_p)$	height of the $keyword_p$ in the syntactic tree of the sentence.

Table 3.3 Feature set for *keyword selection* (potential $keyword$, $keyword_p$ is an element of *keyword-list*)

Term frequency: $term(keyword_p)$ is number of occurrences of the $keyword_p$ in the document. $term(keyword_p)$ is considered as a feature to give preference to the potential *keywords* with high frequency.

In title: $title(keyword_p)$ is a binary feature to check whether $keyword_p$ is present in the title of the document or not. A common word of CS and the title of the document serves as a better keyword for cloze question than the ones that are not present in both.

Height: $height(keyword_p)$ denotes the *height* [3] of the $keyword_p$ in the syntactic tree of the sentence. Height gives an indirect indication of the importance of the word. It also denotes the amount of text in the sentence that modifies the word under consideration.

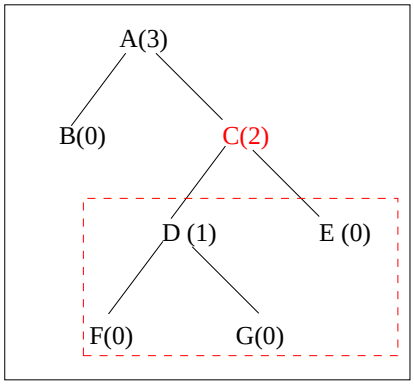

Figure 3.3 Height feature: node (height)

An answerable question should have enough context left after the keyword blanked out. A word with greater *height* in dependency tree gets more score since there is enough context from its dependent words in the syntactic tree to predict the word. For example in Figure 3.3, node *C*'s height is two and the words in the dashed box in its subtree provide the context to answer a question on *C*.

The score of each potential keyword is normalized by the number of words present in it and the best keyword is chosen based on the scores of potential keywords in *keyword-list*. Table 3.4 shows the selected keywords for sample CSs (Table 3.2).

3.3.3 Distractor Selection

Karamanis et al. [21] defines a distractor as, *an appropriate distractor is a concept semantically close to the keyword which, however, cannot serve as the right answer itself.*

For *Distractor Selection*, Brown et al. [9] and Smith et al. [56] used WordNet, Kunichika et al. [24] used their in-house thesauri to retrieve similar or related words (synonyms, hypernyms, hyponyms, antonyms, etc.). However, their approaches can't be used for those domains which don't have ontologies. Moreover, Smith et al. [56] do not select distractors based on the context of the keywords. For

[3]The height of a tree is the length of the path from the deepest node in the tree to the root.

No.	Selected keywords
1	An electron having a certain discrete amount of **energy** is something like a ball on a staircase. *(The Energy Levels of Electrons)*
2	Lipids are the class of large biological molecules that does not include **polymer**. *(Lipids–Diverse Hydrophobic Molecules)*
3	A **DNA** molecule is very long and usually consists of hundreds or thousands of genes. *(Nucleic acids store and transmit hereditary information)*
4	The fatty acid will have a **kink** in its tail wherever a double bond occurs. *(Fats store large amounts of energy)*

Table 3.4 Selected *keywords* for each sample CS

instance, in Example 8 and 9 the keyword *book* occurs in two different senses but same set of distractors will be generated by them.

8. *Book the flight.*

9. *I read a book.*

Feature Symbol	Description
context($distractor_p$, $keyword_s$)	measure of contextual similarity of $distractor_p$ and the $keyword_s$ in which they are present
sim($distractor_p$, $keyword_s$)	*Dice coefficient score* between CS and the sentence containing the $distractor_p$
diff($distractor_p$, $keyword_s$)	difference in *term frequencies* of $distractor_p$ and $keyword_s$ in the chapter

Table 3.5 Feature set for *Distractor Selection* ($keyword_s$ is the selected keyword for a CS, $distractor_p$ is the potential distractor for the $keyword_s$)

So a distractor should come from the same context and domain, and should be relevant. It is also clear from the above discussion that only *term frequency* formula alone will not work for selection of distractors. Our module uses features, shown in Table 3.5, to select three distractors from the set of all potential distractors. Potential distractors are the words in the chapter which have the same POS tag as that of the keyword.

Contextual similarity: *context(*$distractor_p$, $keyword_s$*)* gets the contextual similarity score of a potential distractor and the $keyword_s$ on the basis of context in which they occur in their respective sentences. Value of the feature depends on how similar are the keyword and the potential distractor contextually. The previous two and next two words along with their POS tags are compared to calculate the score.

Sentence Similarity: $sim(distractor_p, keyword_s)$ feature value represents similarity of the sentences in which the $keyword_s$ and the $distractor_p$ occur in. *Dice Coefficient* [11] (equation 3.2) has been used to assign weights to those potential distractors which come from sentences similar to CS because a distractor coming from a similar sentence will be more relevant.

$$dice\ coefficient(s_1, s_2) = \frac{2 \times commontokens}{l(s_1) + l(s_2)} \tag{3.2}$$

Difference in term frequencies: Feature, *diff*($distractor_p$, $keyword_s$) is used to find distractors with comparable importance to the keyword. Term frequency of a word represents its importance in the text and words with comparable importance might be close in their semantic meanings. So, a smaller difference in the term frequencies is preferable.

A word that is present in the CS would not be selected as a distractor. For example in sentence 10, if system selects *oxygen* as a keyword then *hydrogen* will not be considered as a distractor.

10. *Electrons have a negative charge, the unequal sharing of electrons in water causes the* ***oxygen*** *atom to have a partial negative charge and each* ***hydrogen*** *atom a partial positive charge.*

Table 3.6 shows selected three *distractors* for each selected *keywords* (Table 3.4).

keyword	Distractors
energy	charge, mass, water
polymer	acid, glucose, know
DNA	RNA, branch, specific
kink	available, start, method

Table 3.6 Selected distractors for selected keywords, shown in Table 3.4

3.4 Evaluation and Results

Two chapters of the biology book are selected for testing and top 15% candidates are selected by three modules (*Sentence Selection*, *Keyword Selection* and *Distractor Selection*). The modules were manually evaluated independently by two biology students with good English proficiency. Since in current system any kind of post editing or manual work is avoided, comparison of efficiency in manual and automatic generation is not needed unlike Mitkov and Ha et al. [36].

3.4.1 Sentence Selection

The output of the *Sentence Selection* module is a list of sentences. The evaluators check if each of these sentences are good CSs (*informative* and *cloze question-generatable*) or not and binary scoring is done. Evaluators are asked to evaluate selected sentences independently, whether they are useful for learning and answerable, or not.The coverage of the selected sentences w.r.t. the document has not been evaluated.

	Chapter-5	**Chapter-6**	**Total**
No. of Sentences	390	423	813
No. of Selected Sentences	55	65	120
No. of Good CSs (Eval-1)	51	59	110
No. of Good CSs (Eval-2)	44	51	95

Table 3.7 Evaluation of *Sentence Selection*

Evaluator-1 and 2 rated 91.66% and 79.16% of sentences as good potential candidates for cloze question respectively with 0.7 inter evaluator agreement (Cohen's kappa coefficient). Table 3.7 shows the results of *Sentence Selection* for individual chapters. Upon analyzing the bad CSs, we found two different sources of errors. The first source is the feature *first sentence* and the second is lack of used in *Sentence Selection* module.

First sentence: Few documents in the data had either a general statement or a summary of the previous section as the first sentence and the *first sentence* feature contributed to their selection as CS even though they aren't good CSs.

11. *An understanding of energy is as important for students of biology as it is for students of physics, chemistry and engineering.*

For instance, Example 11 isn't a good CS at all even though it occurs as the first sentence in the document.

Less no. of features: Features like *common tokens, superlative and abbreviation, discourse connective at the beginning* and *number of pronouns* was useful in selecting *informative sentences* from the documents. However, in absence of these features in the document, module has selected the CSs on the basis of only two features, *length* and *position of the sentence*. In those cases Evaluators rated few CSs as bad.

12. *Here is another example of how emergent properties result from a specific arrangement of building components.*

For example, sentence 12 rated as a *bad* CS by the evaluators. So more features are need to be to used to avoid this kind of errors.

13. *A molecule has a characteristic **size** and **shape**.*

Apart from these we also found few cases where the context present in the CS wasn't sufficient to answer the question although those sentences were informative. In the Example 13, *size* and *shape* were selected as the keyword that makes cloze question unanswerable because of short context.

3.4.2 Keyword Selection

Our evaluation characterizes a keyword into two categories namely *good* (*G*) and *bad* (*B*). Evaluator-1 and 2 found that 94.16% and 84.16% of the keywords are *good* respectively with inter evaluator agreement 0.75. Table 3.8 shows the results of *Keywords Selection* for individual chapters.

	Chap-5		**Chap-6**		**Total**	
	G	**B**	**G**	**B**	**G**	**B**
Eval-1	50	5	63	2	113	7
Eval-2	50	5	51	14	101	19

Table 3.8 Evaluation of Key(s) Selection: Chap: Chapter, Eval: Evaluator, G and B are for *good* and *bad keyword* respectively

14. *Carbon has a total of **6** electrons , with 2 in the first electron shell and 4 in the second shell.*

We observed that selection of first cardinal as keyword is not always correct. For example, in sentence 14 selection of *6* as the keyword generated trivial CQ.

3.4.3 Distractors Selection

Our system generates four alternatives for each cloze question, out of which three are distractors. To evaluate the distractors' quality, evaluators are asked to substitute the distractor in the gap and check the *readability* and *semantic meaning* of the QS to classify the distractor as *good* or *bad*. Evaluators rate *0, 1, 2* or *3* depending on the number of *good distractors* in the CQ (for example, questions that are rated *2* have two *good distractors* and one *bad distractor*).

15. *An electron having a certain discrete amount of _____ is something like a ball on a staircase.*
*(a) Charge (b) **Energy** (c) Mass (d) Water*
(Class: *3*)

16. *Lipids are the class of large biological molecules that does not include* _____ .
*(a) Acid (b)**Polymer** (c) Glucose (d)* Know
(Class: *2*)

17. *A* _____ *molecule is very long and usually consists of hundreds or thousands of genes.*
(a) ***DNA*** *(b) RNA (c)* Specific *(d)* Branch
(Class: *1*)

18. *The fatty acid will have a* _____ *in its tail wherever a double bond occurs .*
(a) Available *(b)* Method *(c)* ***Kink*** *(d)* Start
(Class: *0*)

Examples of cloze questions generated by our system are shown above (bold alternatives are answers of corresponding questions and italic ones are *bad distractors* and others are *good distractors*).

	Chap-5				**Chap-6**				**Total**			
Class	***0***	***1***	***2***	***3***	***0***	***1***	***2***	***3***	***0***	***1***	***2***	***3***
Eval-1	21	19	12	3	8	31	21	5	29	50	33	8
Eval-2	20	19	13	3	9	25	28	3	29	44	41	6

Table 3.9 Evaluation of *Distractor Selection* (Before any corrections)

Table 3.9 shows the human evaluated results for individual chapter. According to both evaluator-1 and evaluator-2, 75.83% of the cases the system finds *useful cloze questions* with 0.67 inter evaluator agreement. Useful cloze questions are those which have at least one *good distractor*. 60.05% and 67.72% test items are answered correctly by Evaluator 1 and 2 respectively.

We observed that when a keyword has more than one word, distractors' quality reduces because every token in a distractor must be comparably relevant. Small chapter size also effects the number of *good distractors* because distractors are selected from the chapter text.

Syntactic and lexical features are only considered for *Distractor Selection*, the selected distractors could be semantically conflicting with themselves or with the keyword. For example, due to the lack of semantic features in our method a hypernym of the keyword could find way into the distractors list thereby providing a confusing list of distractors to the students. In the example question 1 in section 1, *chemical* which is the hypernym of *covalent* and *ionic* could prove confusing if its one of the choices for the answer. Semantic similarity measures need to be used to solve this problem.

3.5 Comparison

Given the distinct domains in which our system and other systems were deployed, a direct comparison of evaluation scores could be misleading. Hence, in this section we compare our approach with previous approaches in this area.

Smith et al. [56] and Pino et al. [45] used cloze questions for vocabulary learning. Smith et al. [56] present a system, TEDDCLOG, which automatically generates draft test items from a corpus. TEDDCLOG takes the keyword as input. It finds *distractors* from a distributional thesaurus. They got 53.33% (40 out of 75) accuracy after post editing (editing either in carrier sentence (CS) or in *distractors*) in the generated cloze questions.

Pino et al. [45] describe a baseline technique to generate cloze questions (cloze questions) which uses sample sentences from WordNet. They then refine this technique with linguistically motivated features to generate better questions. They used the Cambridge Advanced Learners Dictionary (CALD) which has several sample sentences for each sense of a word for stem selection (CS). The new strategy produced high quality cloze questions 66% of the time.

Karamanis et al. [21] report the results of a pilot study on generating Multiple-Choice Test Items (MCTI) from medical text which builds on the work of Mitkov et al. [38]. Initially keyword set is enlarged with NPs featuring potential keyword terms as their heads and satisfying certain regular expressions. Then sentences having at least one keyword are selected and the terms with the same semantic type in UMLS are selected as *distractors*. In their manual evaluation, the domain experts regarded a MCTI as unusable if it could not be used in a test or required too much revision to do so. The remaining items were considered to be usable and could be post edited by the experts to improve their content and readability or replace inappropriate *distractors*.

They have reported 19% usable items generated from their system and after post editing stems accuracy jumps to 54%. However, our system takes a document and produces a list of CQs by selecting *informative sentences* from the document. It doesn't use any external resources for *Distractors Selection* and finds them in the chapter only that makes it adaptable for those domains which do not have ontologies.

3.6 Summary

Our CQG system, selects most *informative sentences* of the chapters and generates cloze questions on them. Syntactic features helped in quality of cloze questions. We look forward to experimenting on larger data by combining the chapters. Evaluation of course coverage by our system and use of semantic features will be part of our future work.

Chapter 4

Automatic Open-cloze Question Generation System and Evaluation Guidelines

4.1 Overview

In previous chapter we presented a system which generates factual cloze questions from a biology text book through heuristically weighted features. Where we do not use any external knowledge and rely only on information present in the document to generate the CQs with distractors. This restricts the possibilities during distractor selection and leads to low quality distractors. Analysis tell *Distractor Selection* using previous approach lead us low results, because of very small input document size. Finding distractors without any knowledge base is a difficult task.

In this chapter we explain methods to change our previous heuristic steps to rule based techniques but only for first two stages, *Sentence Selection* and *Keywords Selection*. In this chapter we present an automatic open-cloze question generation (OCQG) system. The chapter also include the evaluation guidelines for manual evaluation of cloze-questions.

Open-cloze questions (OCQs) are fill-in-the-blank questions, where a sentence is given with one or more blanks in it and students are asked to fill them. In comparison of cloze questions where four alternatives are given along with question sentence, OCQs are difficult to answer. Also low quality distractors makes question very easy to solve for the students.

1. Question: _____ *was the first Indian batsman to score a double century in an ODI.*
   (a) Sachin (b) Ponting (c) Smith (d) Lara

Example 1 clearly shows that answer of the question must be a name of an Indian batsman. All the distractors except *Sachin* don't belong from Indian Cricket Team, makes question trivial.

2. Sentence: *Riding on their I-League and Federation Cup success, Salgaocar had come into the Durand Cup as one of the favourites.*

Question: *Riding on their _____ and _____ success, Salgaocar had come into the Durand Cup as one of the favorites.*

In above Example 2 *I-League, Federation Cup* is a keyword in the *Sentence1*, so after removing this, an OCQ *Question1* is presented.

Automatic evaluation of a CQG system is a very difficult task; all the previous systems have been evaluated manually. But even for the manual evaluation, one needs specific guidelines to evaluate factual CQs when compared to those that are used in language learning scenario. To the best of our knowledge there are no previously published guidelines for this task.

Cloze questions have one step more than OCQs, which is distractor selection. So there are lot of works for cloze questions but very few for open-cloze specifically. Some of previous systems for OCQs are semi-automatic. For example, [18] and [58] systems take human help somewhere in the process. Either they ask user to select sentences or to select the keywords for the questions, that makes process slow and ineffective. Presented system is fully automatic, generates questions on the content knowledge and not based on heuristically weighted features. Using these guidelines three evaluators report an average score of 3.18 (out of 4) on Cricket World Cup 2011 data.

4.2 Approach

Our system takes news reports on Cricket matches as input and gives factual OCQs as output. Given a document, the system goes through two stages to generate the OCQs. In the first stage, informative and relevant sentences are selected and in the second stage, keywords (or words/phrases to be questioned on) are identified in the selected sentence.

The Stanford CoreNLP tool kit [1] is used for tokenization, POS tagging [61], NER [14], parsing [22] and coreference resolution [27] of sentences in the input documents. There are two different methods to give your input English article to the system, by (i)text and (ii)text file. After collecting the article our system takes two steps to generate questions, (i)Sentence Selection (ii)Keyword Selection.

4.2.1 Sentence Selection

In sentence selection, relevant and informative sentences from a given input article are picked to be the question sentences in cloze questions. [3] uses many summarization features for sentence selection based on heuristic weights. In the task it is difficult to decide the correct relative weights for each feature without any training data. For selection of important sentences our system directly uses a summarizer inspired by [3]. Sometimes summarized sentences are not important for QG, but it seems a good choice in current scenario.

[1] An integrated suite of natural language processing tools for English in Java, including tokenization, part-of-speech tagging, named entity recognition, parsing, and coreference resolution.

There are very few summarizers which produce abstractive (generative) summaries, and they also performed very poorly for example [35]. So our system uses an extractive summarizer, MEAD [2] to select important sentences. Current system takes top 15 percent of the ranked sentences from the summarizer's output.

4.2.2 Keywords Selection

Same as previous chapter, here also *Keyword Selection* is done in two steps, (i) making a list of potential keywords and (ii) pruning of the list to find the best keyword among them. For a good factual OCQ, a keyword should be the word/phrase/clause that tests the knowledge of the user from the content of the article. This keyword shouldn't be too trivial and neither should be too obscure.

Unlike all the previous works our system's questions can have more than one token in a single keyword ([3] (explained in previous chapter) gives more than one token keyword only in case of conjunctions). Having more tokens in a keyword increases the answer scope and judge more knowledge of the students. [3] selected nouns, adjectives and cardinals for their list of potential keywords and then selected the best one based on heuristically weighted features. In our work, a keyword could be a Named Entity (person, number, location, organization or date) (NE), a pronoun or a constituent (selected using the parse tree). For instance, in Example 3, the selected keyword is a noun phrase, *carrom ball*.

3. *R Ashwin used his carrom ball to remove the potentially explosive Kirk Edwards in Cricket World Cup 2011.*

4.2.2.1 Types of keywords

- **Named Entities:** Person or organization's names, numbers, dates, locations, etc are always be a good keywords to ask a questions upon.

4. Sentence: *Steven Finn took a hat-trick as England began their tour of India with a somewhat flattering 56-run victory over a Hyderabad Cricket Association XI.*
   Question1: _____ *took a hat-trick as England began their tour of India with a somewhat flattering 56-run victory over a Hyderabad Cricket Association XI.*
   Question2: *Steven Finn took a hat-trick as England began their tour of India with a somewhat flattering 56-run victory over a* _____.

[2] MEAD is a publicly available toolkit for multi-lingual summarization and evaluation. The toolkit implements multiple summarization algorithms (at arbitrary compression rates) such as position-based, Centroid[RJB00], TF*IDF, and query-based methods (http://www.summarization.com/mead)

In Example 4, from given sentence many questions can be framed based on different NEs as keyword. Out of all two questions (1 and 2) are shown above.

- **Pronouns:** As we stated earlier, it is difficult to generate a sentence to ask an OCQ, which can check deep knowledge about a document of a student. Selection of pronouns is a step towards that aim (checking the deep knowledge). Keeping the accuracy of coreference resolution systems' in mind, we use of those pronouns which come at the beginning of a sentence. Pronoun at the beginning of a sentence make sure that its referent is not present in the sentence.

  5. Sentence: *He is prime minister of India.*
     Question: _____ *is prime minister of India.*

System expects the answer of the above Example 5 is *Mr. Manmohan Singh*, instead of *he*.

- **Constituents:** Using parse trees system taking all possible constituents and push them in the list of potential keywords. Example 6 is given below.

  6. Sentence: *Murray, who has won 21 of his last 22 matches, will wish he could bottle the magic he produced in an astonishing third set, when he dropped just four points.*
     Question: *Murray , who has won 21 of his last 22 matches , will wish he could bottle the magic he produced in an astonishing third set , when he* _____ .

4.2.2.2 Observations

According to our data analysis we have some observations to prune the list that are described below.

- **Relevant tokens should be present in the keyword** There must be few other tokens in a keyword other than stop words[3], common words[4] and topic words [5]. We observed that words given by the TopicS tool are trivial to be keywords as they are easy to predict.

  Many previous system like [3] and [37] used term-frequency is a major feature to select a keyword. But if frequency of a word in high then it would be very easy to answer.

[3] In computing, stop words are words which are filtered out prior to, or after processing of natural language data (text). `http://armandbrahaj.blog.al/2009/04/14/list-of-english-stop-words/`

[4] Most common words in English taken from `http://en.wikipedia.org/wiki/Most\_common\_words\_in\_English`.

[5] Topics (words) which the article talks about. We used the TopicS tool [29]

7. Question: _____ is president of USA.

For instance, if there is an article about Barack Obama and we ask question 7. Isn't it very obvious to answer?

- **Prepositions** The preposition at the beginning of the keyword is an important clue with respect to what the author is looking to check. So, we keep it as a part of the question sentence rather than blank it out as the keyword. We also prune the keywords containing one or more prepositions as they more often than not make the question unanswerable and sometimes introduce a possibility for multiple answers to such questions.

8. Sentence: *England face India in the first of five one-day internationals on Friday.*
   Question1: *England face India in the first* _____.
   Question2: *England face India* _____.

It is clear in above example that if there are more than one prepositions in a keyword then there are multiple answers are possible and it will be difficult to judge students knowledge.

- **A keyword should not start or end at stop word.** We remove the stop words (except articles at start of a keyword) and try to add the new spanned keyword in the list if it is not already present in it.

9. Sentence: *England face India in the first of five one-day internationals on Friday.*
   Question1: England face India in the first of five one-day internationals **on** _____.
   Question2: _____ **in** *the first of five one-day internationals on Friday.*

It is clear from the above Example 9, *Question1* and *Question2* in the Example that is unnecessary to add *on* and *in* in the keywords.

- **Mergeable keywords,** system checks the list and try to merge keywords. If two or more potential keywords are separated by conjunctions then system merger them all and remove the individual potential keywords from the list.

10. Sentence: *Riding on their I-League and Federation Cup success, Salgaocar had come into the Durand Cup as one of the favourites.*
    Question1: *Riding on their I-League and _____ success, Salgaocar had come into the Durand Cup as one of the favorites.*
    Question2: *Riding on their _____ and Federation Cup success, Salgaocar had come into the Durand Cup as one of the favorites.*
    Question3: *Riding on their _____ and _____ success, Salgaocar had come into the Durand Cup as one of the favorites.*

In Example 10, two possible keywords are shown corresponding to *Question1* and *Question2*. But these two can be merged and *Question3* can be generated.

Observations presented by [3] in their keyword selection step, such as, a keyword must not repeat in the sentence again and its term frequency should not be high, a keyword should not be the entire sentence, etc are also used. Scores given by the topicS tool used to filter the keywords with high frequency.

Above criteria reduces the potential keywords' list by a significant amount. Among the rest of the keywords, our system gives preference to NE (persons, location, organization, numbers and dates (in order)), noun phrases, verb phrases in order. To preserve the overall quality of a set of generated questions, system checks that any answer should not be present in other questions. In case of a tie term frequency is used.

4.3 Evaluation Guidelines

Automatic evaluation of any Cloze or Open Cloze QG system is difficult for two reasons i) agreeing on standard evaluation data is difficult ii) there is no one particular set of CQs that is correct. Most question generation systems hence rely on manual evaluation. However, there are no specific guidelines for the manual evaluation either. Evaluation guidelines for a CQG system that we believe are suitable for the task are presented here in Table 4.1. Although open cloze questions do not have distractors, these evaluation guidelines can be used. Because first two steps are common in both type of questions.

Evaluation is done in three phases: (i) Evaluation of selected sentences, (ii) Evaluation of selected keywords and (iii) Evaluation of selected distractors. The evaluation of the selected sentences is done using two metrics, namely, informativeness and relevance. Merging the two metrics into one can mislead because a sentence might be informative but not relevant and vice versa. In such a case, assigning a score of three for one possibility and 2 to the other will not do justice to the system. The keywords are evaluated for their question worthiness and correctness of their span. Finally, the distractors are

Score	Sentence		Keyword	Distractor
4	Very informative	Very relevant	Question worthy	Three are useful
3	Informative	Relevant	Question worthy but span is wrong	Two are useful
2	Remotely informative	Remotely relevant	Question worthy but not the best	One is useful
1	Not at all informative	Not at all relevant	Not at all question worthy	None is useful

Table 4.1 Evaluation Guidelines

evaluated for their usability (i.e. the score is the number of distractors that are useful). A distractor is useful if it can't be discounted easily through simple elimination techniques.

Evaluator		4	3	2	1
Eval-1	Informativeness	8	10	3	1
	Relevance	4	15	3	0
	Keywords	16	0	5	1
Eval-2	Informativeness	13	7	2	0
	Relevance	9	11	2	0
	Keywords	7	0	15	0
Eval-3	Informativeness	9	9	4	0
	Relevance	8	10	4	0
	Keywords	7	0	15	0

Table 4.2 Results (Eval: Evaluator)

It should be noted that the guidelines do a question by question evaluation and the overall performance of the system taking into account the entire document is not performed. This is left for future work. The overall score for every cloze question is calculated by taking the average of all the four metrics for a question. The overall score on the entire data is the mean of scores of each question.

Open-cloze questions generated from news reports on two Cricket World Cup 2011 matches were used for evaluation. 22 questions (10+12) were generated and evaluated by three different evaluators using the above mentioned guidelines. The results are listed in Table 4.2. The overall accuracy of our system is 3.15 (Eval-1), 3.14 (Eval-2) and 3.26 (Eval-3) out of 4.

Chapter 5

Medium Scope Question Generation

5.1 Introduction

Question Generation using discourse connectives means generation of medium scope questions. Medium scope question means, a question having answer of more and one phrase. All the previous work in QG generated questions using a single clause only but to increase the answer scope, we came up with a different method or algorithm which can generate these kind of questions.

Discourse connectives (DC) play a vital role in making the text coherent. They connect two clauses or sentences exhibiting discourse relations such as $temporal$, $causal$, $elaboration$, $contrast$, $result$, etc. Discourse relations have been shown to be useful to generate questions [48] but identifying these relations in the text is a difficult task [46]. So in this work, instead of identifying discourse relations and generating questions using them, we explore the usefulness of discourse connectives for QG. We do this by analyzing the senses of the connectives that help in QG and propose a system that makes use of this analysis to generate questions of the type why, $when$, *give an example* and yes/no.

In this chapter, we present an end-to-end QG system that takes a document as input and outputs all the questions generated using the selected discourse connectives. The system has been evaluated manually by two evaluators for syntactic and semantic correctness of the generated questions. The overall system has been rated 6.3 out of 8 for QGSTEC development dataset and 5.8 out of 8 for Wikipedia dataset.

Question Generation involves two tasks, content selection (the text selected for question generation) and question formation (transformations on the content to get the question). Question formation further has the subtasks of (i) finding suitable question type (wh-word), (ii) auxiliary and main verb transformations and (iii) rearranging the phrases to get the final question.

The system goes through the entire document and identifies the sentences containing at least one of the seven discourse connectives (*because, since, when, although, for example, for instance* and *as a result*). In our approach, suitable *content* for each discourse connective which is referred to as *target argument* is decided based on the properties of discourse connective. The system finds the question type on the basis of discourse relation shown by discourse connective.

5.2 Question type identification for DCs

The sense of the discourse connective influences the question-type (*Q-type*). Since few discourse connectives such as *when, since* and *although* among the selected ones can show multiple senses, the task of sense disambiguation of the connectives is essential for finding the question type.

Since: The connective can show *temporal, causal* or *temporal + causal* relation in a sentence. Sentence exhibits *temporal* relation in presence of keywords like time(7 am), year (1989 or 1980s), start, begin, end, date(9/11), month (January) etc. If the relation is *temporal* then the question-type is *when* whereas in case of *causal* relation it would be *why*.

1. *Single wicket has rarely been played **since** limited overs cricket **began**.*
   Q-type: **when**

2. *Half-court games require less cardiovascular stamina , **since** players need not run back and forth a full court.*
   Q-type: **why**

In Examples 1 and 2, 1 is identified to show *temporal* relation because it has the keyword *began* whereas there is no keyword in the context of Example 2 that gives the hint of *temporal* relation and so the relation here is identified as *causal*.

When: Although *when* shows multiple senses (*temporal, temporal+causal* and *conditional*), we can frame questions by a single question type, *when*. Given a new instance of the connective, finding the correct sense of *when* becomes unnecessary as a result of using discourse connectives.

Although: The connective can show *concession* or *contrast* discourse relations. It is difficult to frame a *wh*-question on *contrast* or *concession* relations. So, system generates a *yes/no* type question for *although*. Moreover, *yes/no* question-type adds to the variety of questions generated by the system.

3. *Greek colonies were not politically controlled by their founding cities , although they often retained religious and commercial links with them .*
   Q-type: **Yes/No**

Identifying the question types for other selected discourse connectives is straight forward because they broadly show only one discourse relation [47]. Based on the relations exhibited by these connectives Q-type for *as a result, for example* and *for instance* will be *why, give an example* and *give an instance*.

5.3 Target arguments for DCs

A discourse connective can realize its two arguments, Arg1 and Arg2, structurally and anaphorically. Arg2 is always realized structurally whereas Arg1 can be either structural or anaphoric [1, 49].

4. *[Arg1 Organisms inherit the characteristics of their parents]* ***because*** *[Arg2 the cells of the offspring contain copies of the genes in their parents' cells.](Intra-sentential connective because)*

5. *[Arg1* The scorers are directed by the hand signals of an umpire.*]* ***For example****, [Arg2 the umpire raises a forefinger to signal that the batsman is out (has been dismissed); he raises both arms above his head if the batsman has hit the ball for six runs.](Inter-sentential connective for example)*

Consider Examples 4 and 5. In 4, Arg1 and Arg2 are the structural arguments of the connective *because* whereas in 5, Arg2 is the structural argument and Arg1 is realized anaphorically. Our system selects one of the two arguments based on the properties of the discourse connectives. Apart from *as a result* for other connectives Arg1 is selected as target argument.

5.4 Question Generation

There are two steps to generate questions using discourse connectives. In the first step target argument is extracted from the dependency parse tree of the sentence and in the next step syntactic transformations are applied to the extracted argument and question is presented.

5.4.1 Target Argument Extraction

Target argument for a discourse connective can be a clause(s) or a sentence(s). It could be one or more sentences in case of *inter-sentential*[1] discourse connectives, whereas one or more clauses in case of *intra-sentential*[2] connectives. Discourse connectives *for example* and *for instance* can realize its Arg1 anywhere in the prior discourse [12]. So the system considers only those sentences in which the connectives occur at the beginning of the sentence and the immediate previous sentence is assumed to be the Arg1 of the connective (which is the *target argument* for QG).

In case of intra-sentential connectives (*because, since, although* and *when*) and *as a result* (*target argument* is Arg2 which would be a clause), identification of *target argument* is done in two steps. The system first locates the syntactic head or head verb of the *target argument* and then extracts it from the dependency tree of the sentence.

[1]Connectives that realize its Arg1 anaphorically and Arg2 structurally
[2]Connectives that realize both of its arguments structurally

Approach for locating the syntactic head of *target argument* is explained with the help of Figure 5.1. Syntactic head of Arg2 is the first finite verb while percolating up in the dependency tree starting from the discourse connective. In case of intra-sentential connectives where Arg1 is the *target argument*, the system percolates up until it gets the second finite verb which is assumed to be target head of Arg1. Number of percolations entirely depend on structure and complexity of the sentence.

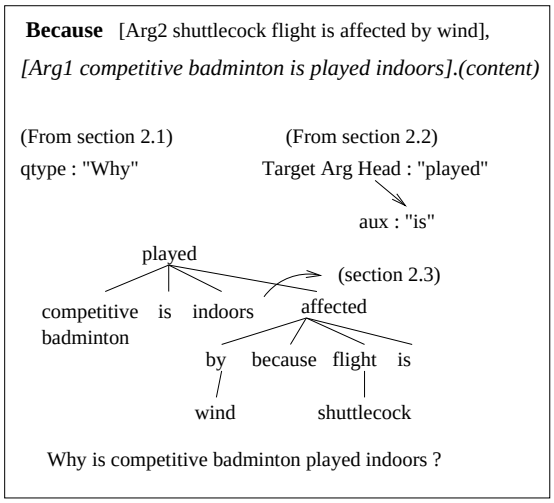

Figure 5.1 Question Generation process

Since the discourse connective in the example of Figure 5.1 is *because*, the *target argument* is Arg1. By percolating up the tree starting from *because*, the head of Arg2 is *affected* and that of Arg1 is *played*. Once we locate the head of the *target argument*, we find the auxiliary as [32] does. For the example in Figure 5.1, the auxiliary for question generation is *is*.

The extraction of the *target argument* is done after identifying its syntactic head. For *as a result*, the *target argument*, Arg2, is the subtree with head as the head of the connective. For intra-sentential connectives, the *target argument*, Arg1, is the tree remaining after removing the subtree that contains Arg2.

In Figure 5.1, after removing the unwanted argument Arg2 (subtree with head *affected*), the system gets *competitive badminton is played indoors* which is the required clause (*content*) for question generation. The next section describes how the *content* is transformed into a question.

5.4.2 Syntactic Transformations

The syntactic transformations used in this work are similar to those by [32]. At this stage, the system has the question type, auxiliary and the *content*. The following set of transformations are applied on the *content* to get the final question. (1) If the auxiliary is present in the sentence itself then it is moved to the beginning of the sentence; otherwise auxiliary is added at the beginning of the sentence. (2) If

a wh-question is to be formed, the question word is added just before the auxiliary. In case of Yes/No questions, the question starts with the auxiliary itself as no question word is needed. (3) A question-mark(?) is added at the end to complete the question.

Consider the example in Figure 5.1. Here the *content* is *competitive badminton is played indoors*. Applying the transformations, the auxiliary is first moved at the start of the sentence to get *is competitive badminton played indoors*. Then the question type *Why* is added just before the auxiliary *is*, and a question-mark is added at the end to get the final question, *Why is competitive badminton played indoors ?*

5.5 Evaluation and Results

Automatic evaluation of any natural language generated text is difficult. So, our system is evaluated manually. The evaluation was performed by two graduate students with good English proficiency. Evaluators were asked to rate the questions on the scale of 1 to 4 (4 being the best score) on syntactic and semantic correctness [16] of the question and an overall rating on the scale of 8 (4+4) is assigned to each question.

The overall system is rated 6.3 out of 8 on QGSTEC-2010 development dataset and the total number of questions generated for this dataset is 61. The instances of the connectives were less in this dataset. So, the system is further tested on five Wikipedia articles (football, cricket, basketball, badminton and tennis) for effective evaluation. Results. Overall rating of the system is 5.8 out of 8 for this dataset and 150 are the total number of questions generated for this dataset. The inter-evaluator agreement (Cohen's kappa coefficient) for the QGSTEC-2010 development dataset for syntactic correctness measure is 0.6 and is 0.5 for semantic correctness measure, and in case of Wikipedia articles the agreement is 0.7 and 0.6 for syntactic and semantic correctness measures respectively.

On analyzing the data, we found that the Wikipedia articles have more complex sentences (with unusual structure as well as more number of clauses) than QGSTEC-2010 development dataset. As a result, the system's performance consistently drops for all the connectives in case of Wikipedia dataset.

5.6 Error Analysis

An error analysis was carried out on the system's output and the four most frequent types of errors are listed in this section.

- Coreference resolution, Since the system doesn't handle CR, some questions are left with pronouns in them and evaluators gave lesser semantic scores.

- Parsing Errors, Sometimes the parser fails to give a correct parse for the sentences with complex structure. In such cases, the system generates a question that is unacceptable.

- Errors due to the inter-sentential connectives, our assumption for the inter-sentential connectives is not always correct. So that assumption is also lead us to wrong questions sometimes.
- Fluency issues, The system does not handle the removal of predicative adjuncts. So the questions with optional phrases in it are rated low for syntactic correctness measure.

Chapter 6

Conclusion and Future Work

6.1 Summary

This section summarizes the chapters have been presented so far.

Chapter-3 provided an algorithm to generate cloze questions heuristically without using any external resources. Generation of cloze questions involve three stages. (i) *Sentence Selection*, (ii) *Keyword Selection* and (iii) *Distrator Selection*. Our proposed method selects number of feature for each stage and based on the heuristic weights system finds the potential candidate in each stage. In sentence selection system mainly used summarization features, for keywords selection, it selects only nouns, adjectives and cardinals. In the third stage of distractor selection, apart from POS tag of the selected keyword, system also uses a ranking method for all possible potential distractors and present the best three. Finally system presents a question sentence (a sentence with one or more blanks) and 4 alternatives. For evaluation of this system, we used manual evaluators. System performed well enough on small size input datasets.

In chapter-4 we tried to improve our previous model of cloze question generation (presented in chapter-3). Basically the previous system was heuristic feature based, this system changed first two stages, *Sentence Selection* and *Keyword Selection* using a rule based model. So the chapter presents an open-cloze questions generation system. Here in the sentence selection a summarizer is used directly and for keyword selection first all NEs, constituents and pronouns are selected for generating potential keywords list. Then pruning is done to select one best keyword. Here also evaluation is done manually.

In chapter-5, we presented a method for automatic generation of questions using discourse connectives. Since a connective connect two clauses using its sense, the sense could be used to generate medium scope questions. The method generates different wh-type questions using discourse connectives. It also includes sense disambiguation for few discourse connectives, like *since* and *when*. Out of two clauses, method uses one for question framing and other for answer scope of the generated question. Depending on the discourse relation system decides the question-type of the question. Finally our method generates various type of medium answer scope questions.

6.2 Contribution

Most of the previous works in area of *Question Generation* generated questions for language learning. There are very few works which generate factual questions, and there is no work which generate cloze-questions on factual knowledge from a document. Our works generates factual questions from a document without using any external resources. All the works on cloze-questions have used external resources for *Distractor Selection,* which makes them domain dependent and difficult to use for non technical persons. Features used in our system makes it adaptable for other languages too. Ours is the first one which generates medium scope factual questions from discourse connectives. We made use of discourse connectives to generate medium scope questions. So overall this work gives a method to generated factual cloze-questions without using any external resources and generation of medium scope questions from a document. That makes our system unique and useful too.

6.3 Future Work

6.3.1 Improving Distractor Selection

Mainly our cloze-question generation system has three stages, out of those *Sentence Selection* and *Keyword Selection* work well. But the third stage *Distractor Selection* didn't work very well. As also said by the evaluation, it gets average score of between *1* and *2*. That score means on an average system finds *1* or *2 good distractors* out of *3*. So every time either *1* or *2* alternatives are ruled out very easily. But for a method which doesn't use any external resources we can't expect much. Although we can use some domain specific knowledge base in our third stage, *Distractor Selection*. *wordnet* , *web* can also be used for the task in future.

6.3.2 Learning methods for QG

We will make our system available online so user can give their feedbacks corresponding to each selected sentence, keyword and distractor. By collecting the feedbacks we plan to build a training dataset which is not available for the task yet. So in future one can make use of the annotated data and apply some learning algorithm to generate good quality of questions.

6.3.3 Adding more Discourse connectives in medium scope question generation

Our approach of generating medium scope questions using discourse connectives uses seven discourse connectives. Those seven discourse connectives involves *since, because, when, while, for example, for instance* and *although*. As we discussed in chapter 5 these connectives almost covers 42% of all other connectives. So there are some connectives (for example *however*) left on which medium scope

question can be generated. Also our system didn't perform well enough in case of *when* connective, there is also some scope is left for future work.

Also apart from discourse connectives we can use different phrases like, *not only but also, either-or* and *neither-nor* for the same. From these and all others which follow some kind of pattern in their use, a good quality question can be generated out of them.

6.3.4 On Different Language

We have achieved good accuracies in QG for specific and medium scope questions. Since the methods used in the generation process are independent of the language, a similar system can be built on the same lines as this system for other languages like, Hindi, Spanish, etc.

Related Publications

- **Manish Agarwal** and Prashanth Mannem. Automatic Gap-fill Questions Generation from Text-Books. In the Proceedings of The 6th Workshop on Innovative Use of NLP for Building Educational Applications, ACL-HLT 2011.
- **Manish Agarwal**, Rakshit Shah and Prashanth Mannem. Automatic Questions Generation using Discourse Cues. In the Proceedings of The 6th Workshop on Innovative Use of NLP for Building Educational Applications, ACL-HLT 2011.

Bibliography

[1] Pdtb 2.0 annotation manual. In *http://www.seas.upenn.edu/ pdtb/PDTBAPI/pdtb-annotation-manuat.,pdf*, 2007.

[2] Question generation shared task and evaluation challenge. In *http://questiongeneration.org/QG2010*, 2010.

[3] M. Agarwal and P. Mannem. Automatic gap-fill questions generation from text-books. In *In the Proceedings of The 6th Workshop on Innovative Use of NLP for Building Educational Applications, ACL-HLT 2011*, 2011.

[4] M. Agarwal, R. Shah, and P. Mannem. Automatic questions generation using discourse cues. In *In the Proceedings of The 6th Workshop on Innovative Use of NLP for Building Educational Applications, ACL-HLT 2011*, 2011.

[5] Ainsworth and S. Evaluating the redeem authoring tool: Can teachers create effective learning environments? In *International Journal of Artificial Intelligence in Education, 14(3)*, 2004.

[6] T. Aleahmad, V. Aleven, and R. Kraut. Open community authoring of targeted worked example problems. In *In Proc. of Intelligent Tutoring Systems*, 2008.

[7] J. R. Bormuth. Cloze readability procedure. In *University of California, Los Angeles. CSEIP Occasional Report No. 1*, 1967.

[8] K. E. Boyer, E. Y. Ha, M. D. Wallis, R. Phillips, M. A. Vouk, and J. C. Lester. Discovering tutorial dialogue strategies with hidden markov models. In *In Proc. of AIED*, 2009.

[9] G. A. Brown, Jonathan C. andFrishkoff and M. Eskenazi. Automatic question generation for vocabulary assessment. In *Proc. of HLT/EMNLP '05, pp. 819-826*, 2005.

[10] J. C. BROWN, G. A. FRISHKOFF, and M. ESKENAZI. Automatic question generation for vocabulary assessment. In *In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, British Columbia, Canada*, 2005.

[11] L. R. Dice. Measures of the amount of ecologic association between species. 1945.

[12] R. Elwell and J. Baldridge. Discourse connective argument identification with connective specific rankers. In *In Proceedings of ICSC-2008*, 2008.

[13] M. Feng and N. T. Heffernan. Informing teachers live about student learning: Reporting in the assistment system. In *Technology, Instruction, Cognition, and Learning, 3*, 2006.

[14] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 363-370*, 2005.

[15] A. C. GRAESSER, L. K. VAN, C. P. ROSE, P. W. JORDAN, and D. HARTER. Intelligent tutoring systems with conversational dialogue. In *AI Magazine, 22(4), 3952*, 2001.

[16] E. guidelines. In qgstec-2010 task b evaluation guidelines. In *http://www.question generation.org/QGSTEC2010/ uploads/QG-fromSentences-v2.doc*, 2010.

[17] M. Heilman, L. Zhao, J. Pino, and M. Eskenazi. Retrieval of reading materials for vocabulary and reading practice. In *In Proc. of the Third Workshop on Innovative Use of NLP for Building Educational Applications*, 2008.

[18] A. HOSHINO and H. NAKAGAWA. A cloze test authoring system and its automation. In *In Proceeding of ICWL2007*, 2007.

[19] A. Husam, C. Yllias, and A. H. Sadid. Automation of question generation from sentences. In *Proceedings of QG2010: The Third Workshop on Question Generation*, 2010.

[20] M. C. G. O. J., M. O. J., J. O. J., and F. S. A web-based testing system with dynamic question generation. In *In ASEE/IEEE Frontiers in Education Conference*, 2001.

[21] N. Karamanis, L. A. Ha, and R. Mitkov. Generating multiple-choice test iterms from medical text: A pilot study. In *In Proceedings of INLG 2006, Sydney, Australia*, 2006.

[22] D. Klein and D. M. Christopher. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423-430*, 2003.

[23] K. R. Koedinger, J. R. Anderson, W. H. Hadley, and M. A. Mark. Intelligent tutoring goes to school in the big city. In *International Journal of Artificial Intelligence in Education, 8*, 1997.

[24] H. Kunichika, M. Urushima, T. Hirashima, and A. Takeuchi. A computational method of complexity of questions on contents of english sentences and its evaluation. In *In: Proc. of ICCE 2002, Auckland, NZ, pp. 97101 (2002)*, 2002.

[25] T. W. LAUER, E. PEACOCK, and A. C. GRAESSER. Questions and information systems. 1992.

[26] C. Leacock, M. Chodorow, M. Gamon, and J. Tetreault. Automated grammatical error detection for language learners. In *Morgan and Claypool*, 2010.

[27] H. Lee, Y. Peirsman, A. Chang, N. Chambers, M. Surdeanu, and D. Jurafsky. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *In Proceedings of the CoNLL-2011 Shared Task, 2011*, 2011.

[28] J. Lee and S. Seneff. Automatic generation of cloze items for prepositions. In *CiteSeerX - Scientific Literature Digital Library and Search Engine [http://citeseerx.ist.psu.edu/oai2] (United States)*, 2007.

[29] C.-Y. Lin and E. Hovy. The automated acquisition of topic signatures for text summarization. In *In Proceedings of COLING 2000*, 2000.

[30] Y. C. Lin, C. Sung, L. C., and M. C. An automatic multiple-choice question generation scheme for english adjective understanding. In *CCE 2007 Workshop Proc. of Modeling, Management and Generation of Problems / Questions in eLearning, pp. 137-142*, 2007.

[31] D. Litman and S. Silliman. Itspoke: An intelligent tutoring spoken dialogue system. In *In Companion Proc. of HLT/NAACL*, 2004.

[32] P. Mannem, R. Prasad, and A. Joshi. Question generation from paragraphs at upenn: Qgstec system discription. In *Proceedings of QG2010: The Third Workshop on Question Generation*, 2010.

[33] B. M. McLaren, R. Wegerif, J. Mikatko, O. Scheuer, M. Chamrada, and N. Mansour. Are your students working creatively together? automatically recognizing creative turns in student e-discussions. In *In Proc. of AIED*, 2009.

[34] D. Meurers, R. Ziai, L. Amaral, A. Boyd, A. Dimitrov, V. Metcalf, and N. Ott. Enhancing authentic web pages for language learners. In *In Proc. of the 5th Workshop on Innovative Use of NLP for Building Educational Applications*, 2010.

[35] J. Michael and O. M. Vibhu. Ultra-summarization: A statistical approach to generating highly condensed non-extractive summaries. In *In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999.

[36] R. Mitkov and L. A. Ha. Computer-aided generation of mukoedinger, k. r., anderson, j. r., hadley, w. h., and mark, m. a. (1997). intelligent tutoring goes to school in the big city. international journal of artificial intelligence in education, 8.ltiple-choice tests. In *Proceedings of the HLT/NAACL 2003 Workshop on Building educational applications using Natural Language Processing. Edmonton, Canada, 17-22.*, 2003.

[37] R. Mitkov and L. A. Ha. Computer-aided generation of multiple-choice tests. In *In Proceedings of Workshop On Building Educational Applications Using Natural Language Processing*, 2003.

[38] R. Mitkov, L. A. Ha, and N. Karamanis. A computer-aided environment for generating multiple-choice test items. In *Natural Language Engineering 12(2): 177-194*, 2006.

[39] M. Mohler and R. Mihalcea. Text-to-text semantic similarity for automatic short answer grading. In *In Proc. of EACL*, 2009.

[40] J. Mostow, J. Beck, J. Bey, A. Cuneo, J. Sison, B. Tobin, and J. Valeri. Using automated questions to assess reading comprehension, vocabulary, and effects of tutorial interventions. In *Technology, Instruction, Cognition and Learning, 2:97134*, 2004.

[41] A. NDRENUCCI and S. NEIDERS E. Automated question answering : Review of the main approaches. In *In Proceedings of the 3rd International Conference on Information Technology and Applications (ICITA05), Sydney, Australia*, 2005.

[42] R. D. Nielsen, W. Ward, J. H. Martin, and M. Palmer. Extracting a representation from text for semantic analysis. In *In Proc. of ACL-08:HLT*, 2008.

[43] J. Oller and W. J. Scoring methods and difficulty levels for cloze tests of proficiency in english as a second language. In *The Modern Language Journal, 56(3)*, 1972.

[44] S. Paland, T. Mondal, P. Pakray, D. Das, and S. Bandyopadhyay. Qgstec system description juqgg: A rule based approach. In *Proceedings of QG2010: The Third Workshop on Question Generation*, 2010.

[45] J. Pino, M. Heilman, and M. Eskenazi. A selection strategy to improve cloze question quality. In *Wkshop on Intelligent Tutoring Systems for Ill-Defined Domains. 9th Int. Conf. on ITS*, 2009.

[46] E. Pitler, A. Louis, and A. Nenkova. Automatic sense prediction for implicit discourse relations in text. In *ACL '09 Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2.*, 2009.

[47] E. Pitler and A. Nenkova. Using syntax to disambiguate explicit discourse connectives in text. In *ACLShort '09 Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, 2009.

[48] R. Prasad and A. Joshi. A discourse-based approach to generating why-questions from text. In *Proceedings of the Workshop on the Question Generation Shared Task and Evaluation Challenge Arlington, VA, September 2008*, 2008.

[49] R. Prasad, A. Joshi, and B. Webber. Exploiting scope for shallow discourse parsing. In *LREC2010*, 2010.

[50] E. F. Rankin and J. W. Culhane. Comparable cloze and multiple-choice comprehension test scores. In *Journal of Reading, 13(3)*, 1969.

[51] L. Razzaq, J. Patvarczki, Almeida, S. F., Vartak, M. M., Feng, N. T. Heffernan, and K. R. Koedinger. The assistment builder: Supporting the life cycle of tutoring system content creation. In *IEEE Transactions on Learning Technologies, 2(2)*, 2009.

[52] S. Ritter. The authoring assistant.. In *In Proc. of Intelligent Tutoring Systems*, 1998.

[53] V. RUS and A. C. GRAESSER. The question generation shared task and evaluation challenge. In *In Workshop on the Question Generation Shared Task and Evaluation Challenge, Final Report, The University of Memphis : National Science Foundation*, 209.

[54] M. D. Shermis and J. Burstein. Automated essay scoring: A cross-disciplinary perspective. In *MIT Press*, 2003.

[55] H. J. Smith, Higgins, S., K. Wall, and J. Miller. Interactive whiteboards: boon or bandwagon? a critical review of the literature. In *Journal of Computer Assisted Learning*, 2005.

[56] S. Smith, P. Avinesh, and A. Kilgarriff. Gap-fill tests for language learners: Corpus-driven item generation. 2010.

[57] E. Sneiders. Automated question answering using question templates that cover the conceptual model of the database. In *In Proceedings of the 6th International Conference on Applications of Natural Language to Information Systems (pp. 235-239)*, 2002.

[58] L. Stanescu, C. S. Spahiu, A. Ion, and A. Spahiu. Question generation for learning evaluation. In *In Proceeding of IMCSIT 2008*, 2008.

[59] E. Sumita, F. Sugaya, and S. Yamamoto. Measuring non-native speakers proficiency of english by using a test with automatically-generated fill-in-the-blank questions. In *2nd Wkshop on Building Educational Applications using NLP, Ann Arbor*, 2005.

[60] A. Szabo and N. Hastings. Using it in the undergraduate classroom: should we replace the blackboard with powerpoint? In *Computers and Education, 35(3)*, 2000.

[61] K. Toutanova, D. Klein, C. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *In Proceedings of HLT-NAACL 2003, pp. 252-259*, 2003.

[62] A. R. Trees and M. H. Jackson. The learning environment in clicker classrooms: student processes of learning and involvement in large university-level courses using student response systems. In *Learning, Media and Technology, 32(1)*, 2007.

[63] K. Vanlehn, C. Lynch, K. Schulze, J. A. Shapiro, R. Shelby, L. Taylor, D. Treacy, A. Weinstein, and M. Wintersgill. The andes physics tutoring system: Lessons learned. In *International Journal of Artificial Intelligence in Education, 15(3)*, 2005.

[64] A. Varga and L. An Ha. Wlv: A question generation system for the qgstec 2010 task b. In *Proceedings of QG2010: The Third Workshop on Question Generation*, 2010.

[65] W. WANG, H. TIANYONG, and L. WENYIN. Automatic question generation for learning evaluation in medicine. In *In LNCS Volume 4823*, 2008.

[66] B. P. Woolf. Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning. In *Morgan Kaufmann*, 2008.