Automatic Gap-fill Question Generation from Text Books

Manish Agarwal and Prashanth Mannem
Language Technologies Research Center
International Institute of Information Technology
Hyderabad, AP, India - 500032
{manish.agarwal,prashanth}@research.iiit.ac.in

Abstract
In this paper, we present an automatic question generation system that can generate gap-fill questions for content in a document. Gap-fill questions are fill-in-the-blank questions with multiple choices (one correct answer and three distractors) provided. The system finds the informative sentences from the document and generates gap-fill questions from them by first blanking keys from the sentences and then determining the distractors for these keys. Syntactic and lexical features are used in this process without relying on any external resource apart from the information in the document. We evaluated our system on two chapters of a standard biology textbook and presented the results.

1 Introduction
Gap-fill questions are fill-in-the-blank questions, where one or more words are removed from a sentence/paragraph and potential answers are listed. These questions, being multiple choice ones, are easy to evaluate. Preparing these questions manually will take a lot of time and effort. This is where automatic gap-fill question generation (GFQG) from a given text is useful.

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 gap-fill question (GFQ) such as the one above, 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 gap-fill sentence (GFS). The word(s) which is removed from a GFS to form the QS is referred to as the key while the three alternatives in the question are called as distractors, as they are used to distract the students from the correct answer.

Previous works in GFQG (Sumita et al., 2005; John Lee and Stephanie Seneff, 2007; Lin et al., 2007; Pino et al., 2009; Smith et al., 2010) have mostly worked in the domain of English language learning. Gap-fill questions have been generated to test student’s knowledge of English in using the correct verbs (Sumita et al., 2005), prepositions (John Lee and Stephanie Seneff, 2007) and adjectives (Lin et al., 2007) in sentences. Pino et al. (2009) and Smith et al. (2010) have generated GFQs to teach and evaluate student’s vocabulary.

In this paper, we move away from the domain of English language learning and work on generating gap-fill 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 gap-fill 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 (GFSs) are generated by first choosing a key in each of these and then finding appropriate distractors for them from the chapter.

Our GFQG system takes a document with its title as an input and produces a list of gap-fill questions as

\(^1\)A sentence is deemed informative if it has the relevant course knowledge which can be questioned.
output. Unlike previous works (Brown et al., 2005; Smith et al., 2010) it doesn’t use any external resource 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 gap-fill questions but also for students who need an automatic question generator to aid their learning from a textbook.

2 Data Used
A Biology text book *Campbell Biology, 6th Edition* has been used for work in this paper. We have reported results of our system on 2 chapters (*the structure and function of macromolecules* and *an introduction to metabolism*) of unit 1. Each chapter contains sections and subsections with their respective topic headings. Number of subsections, sentences, 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 GFQG independently.

3 Approach
Given a document, the gap-fill questions are generated from it in three stages: sentence selection, key selection and distractor selection. *Sentence selection* involves identifying informative sentences in the document which can be used to generate a gap-fill question. These sentences are then processed in the *key selection* stage to identify the key on which to ask the question. In the final stage, the *distractors* for the selected key are identified from the given chapter by searching for words with the same context as that of the key.

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 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.

\[ \text{score} = \sum_{i=0}^{n} w_i \times f_i \]  

(1)

In equation 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 1.

Figure 1: System architecture

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

3.1 Sentence Selection
A good GFS should be (1) informative and (2) gap-fill 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 gap-fill question-generatable if there is sufficient context within the sentence to predict the key 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 1 gives a summary of the features used.

**First sentence:** \( f(s_1) \) is a binary feature to check whether the sentence \( s_1 \) is the first sentence of the document or not. Upon analysing the documents in the textbook, it was observed that the first sentence in the document usually provides a summary of the document. Hence, \( f(s_1) \) has been used to make use of the summarized first sentence of the document.
Table 1: Feature set for Sentence Selection ($s_i$: $i^{th}$ sentence of the document; $G$: to capture informative sentences; $I$: to capture the potential candidate for generating a GFQs)

<table>
<thead>
<tr>
<th>Feature Symbol</th>
<th>Description</th>
<th>Criterion</th>
</tr>
</thead>
<tbody>
<tr>
<td>discon($s_i$)</td>
<td>$s_i$ beginning with a discourse connective?</td>
<td>G</td>
</tr>
<tr>
<td>l($s_i$)</td>
<td>Number of words in $s_i$</td>
<td>G</td>
</tr>
<tr>
<td>noun($s_i$)</td>
<td>No. of nouns in $s_i$ / length($s_i$)</td>
<td>G</td>
</tr>
<tr>
<td>pronoun($s_i$)</td>
<td>No. of pronouns in $s_i$ / length($s_i$)</td>
<td>G</td>
</tr>
<tr>
<td>super($s_i$)</td>
<td>Does $s_i$ contain a word in its superlative degree?</td>
<td>I</td>
</tr>
<tr>
<td>abbrev($s_i$)</td>
<td>Does $s_i$ contain any abbreviation?</td>
<td>G</td>
</tr>
<tr>
<td>sim($s_i$)</td>
<td>No. of tokens common in $s_i$ and title / length($s_i$)</td>
<td>I</td>
</tr>
<tr>
<td>$f(s_i)$</td>
<td>Is $s_i$ the first sentence of the document?</td>
<td>I</td>
</tr>
</tbody>
</table>

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 key.

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)

For example sentence 2, value of the feature is 3/19 (common words:3, sentence length:19) and generating gap-fill question using energy, levels or electrons as the key will be useful.

Abbreviations and Superlatives: abbrev($s_i$), 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.

For example, in sentence 3, presence of strongest makes sentence more informative and useful for generating a gap-fill question.

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 GFSs 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 connective2 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 key, 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 noun($s_i$) and pronoun($s_i$) 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 keys to ask a gap-fill question on.

Four sample GFSs are shown in Table 3 with their document’s titles.

3.2 Key Selection

For each sentence selected in the previous stage, the key selection stage identifies the most appropriate key from the sentence to ask the question on.

Previous works in this area, Smith et al. (2010) take keys as an input and, Karamanis et al. (2006) and Mitkov et al. (2006) select keys on the basis of term frequency and regular expressions on nouns. Then they search for sentences which contain that particular key in it. Since their approaches generate gap-fill questions only with one blank, they could end up with a trivial GFQ, especially in case of conjunctions.

2because, since, when, thus, however, although, for example and for instance connectives have been included.
5. Somewhere in the transition from molecules to cells, we will cross the blurry boundary between nonlife and life.

For instance in example sentence 5, selecting only one of non-life and life makes the question trivial. This is another reason for performing sentence selection before key selection. Our system can generate GFQs with multiple blanks unlike previous works described above.

Our approach of key selection from a GFS is a two-step process. In the first step the module generates a list of potential keys from the GFS (key-list) and in the second step it selects the best key from this key-list.

3.2.1 Key-list formation

A list of potential keys 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 key-list. In figure 2(A), the three noun chunks the strongest kind, chemical bond and covalent bond and ionic bond are pushed into the key-list.

2. For each sequence in the key-list, the most important word(s) is selected as the potential key and the other words are removed. The most important word in a noun chunk in the context of GFQG 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 key. If the noun chunk is a NP coordination, both the conjuncts are selected as a single potential key making it a case of multiple gaps in QS. In Figure 2(B) potential keys 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 key 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 keys 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 key-list which occurs more than once in the GFS is discarded as a potential key as it more often than not generates a trivial question. For example, in sentence 6 selecting any one of the two electron as a key generates an easy gap-fill 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 GFS, the first one is chosen as its key directly and a gap-fill question has been generated (see example 7).

3.2.2 Best Key selection

In this step three features, term(key_p), title(key_p) and height(key_p), described in Table 2, are used to select the best key from the key-list.

<table>
<thead>
<tr>
<th>Feature Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>term(key_p)</td>
<td>Number of occurrences of the key_p in the document.</td>
</tr>
<tr>
<td>title(key_p)</td>
<td>Does title contain key_p?</td>
</tr>
<tr>
<td>height(key_p)</td>
<td>height of the key_p in the syntactic tree of the sentence.</td>
</tr>
</tbody>
</table>

Table 2: Feature set for key selection (potential key, key_p is an element of key-list)

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

**In title:** title(key_p) is a binary feature to check whether key_p is present in the title of the document or not. A common word of GFS and the title of the document serves as a better key for gap-fill question than the ones that are not present in both.

**Height:** height(key_p) denotes the height of the key_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:** Height feature: node (height)

An answerable question should have enough context left after the key blanked out. A word with greater height in a 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, 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 key is normalized by the number of words present in it and the best key is chosen based on the scores of potential keys in key-list. Table 3 shows the selected keys (red colored) for sample GFSs.

### 3.3 Distractor Selection

Karamanis et al. (2006) defines a distractor as, an appropriate distractor is a concept semantically close to the key which, however, cannot serve as the right answer itself.

For distractor selection, Brown et al. (2005) and Smith et al. (2010) used WordNet, Kunichika et al. (2002) 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. (2010) do not select distractors based on the context of the keys. For example, in the sentences 8 and 9, the key book occurs in two different senses but same set of distractors will be generated by them.

8. Book the flight.


### Table 3: Selected keys for each sample GFS

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

### Table 4: Feature set for distractor selection (key_s is the selected key for a GFS, distractor_p is the potential distractor for the key_s)

<table>
<thead>
<tr>
<th>Feature Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>context(distractor_p, key_s)</td>
<td>measure of contextual similarity of distractor_p and the key_s in which they are present</td>
</tr>
<tr>
<td>sim(distractor_p, key_s)</td>
<td>Dice coefficient score between GFS and the sentence containing the distractor_p</td>
</tr>
<tr>
<td>diff(distractor_p, key_s)</td>
<td>difference in term frequencies of distractor_p and key_s in the chapter</td>
</tr>
</tbody>
</table>

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 4, 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 key.
Contextual similarity: \( \text{context}(\text{distractor}_p, \text{key}_s) \) gets the contextual similarity score of a potential distractor and the keys on the basis of context in which they occur in their respective sentences. Value of the feature depends on how similar are the key 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: \( \text{sim}(\text{distractor}_p, \text{key}_s) \) feature value represents similarity of the sentences in which the keys and the distractor occur in. Dice Coefficient (Dice, 1945) (equation 2) has been used to assign weights to those potential distractors which come from sentences similar to GFS because a distractor coming from a similar sentence will be more relevant.

\[
\text{dice coefficient}(s_1, s_2) = \frac{2 \times \text{common tokens}}{l(s_1) + l(s_2)} \tag{2}
\]

Difference in term frequencies: Feature, \( \text{diff}(\text{distractor}_p, \text{key}_s) \) is used to find distractors with comparable importance to the key. 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.

<table>
<thead>
<tr>
<th>key</th>
<th>Distractors</th>
</tr>
</thead>
<tbody>
<tr>
<td>energy</td>
<td>charge, mass, water</td>
</tr>
<tr>
<td>polymer</td>
<td>acid, glucose, know</td>
</tr>
<tr>
<td>DNA</td>
<td>RNA, branch, specific</td>
</tr>
<tr>
<td>kink</td>
<td>available, start, method</td>
</tr>
</tbody>
</table>

Table 5: Selected distractors for selected keys, shown in Table 3

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.

A word that is present in the GFS would not be selected as a distractor. For example in sentence 10, if system selects oxygen as a key then hydrogen will not be considered as a distractor. Table 5 shows selected three distractors for each selected keys.

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, key 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. (2003).

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 GFSs (informative and gap-fill 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.

<table>
<thead>
<tr>
<th></th>
<th>Chapter-5</th>
<th>Chapter-6</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Sentences</td>
<td>390</td>
<td>423</td>
<td>813</td>
</tr>
<tr>
<td>No. of Selected Sentences</td>
<td>55</td>
<td>65</td>
<td>120</td>
</tr>
<tr>
<td>No. of Good GFSs (Eval-1)</td>
<td>51</td>
<td>59</td>
<td>110</td>
</tr>
<tr>
<td>No. of Good GFSs (Eval-2)</td>
<td>44</td>
<td>51</td>
<td>95</td>
</tr>
</tbody>
</table>

Table 6: Evaluation of Sentence Selection

Evaluator-1 and 2 rated 91.66% and 79.16% of sentences as good potential candidates for gap-fill question respectively with 0.7 inter evaluator agreement (Cohen’s kappa coefficient). Table 6 shows the results of sentence selection for individual chapters. Upon analysing the bad GFSs, 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 GFS even though they aren’t good GFSs.

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

For example, the system generated a gap-fill
question on example 11 which isn’t a good GFS 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 GFSs on the basis of only two features, length and position of the sentence. In those cases Evaluators rated few GFSs 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 GFS by the evaluators. So more features are need to be 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 GFS wasn’t sufficient to answer the question although those sentences were informative. In the above example 13, *size and shape* were selected as the key that makes gap-fill question unanswerable because of short context.

**4.2 Key Selection**

Our evaluation characterizes a key into two categories namely good (G) and bad (B). Evaluator-1 and 2 found that 94.16% and 84.16% of the keys are good respectively with inter evaluator agreement 0.75. Table 7 shows the results of keys selection for individual chapters.

<table>
<thead>
<tr>
<th>Chap-5</th>
<th>Chap-6</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>G</td>
<td>B</td>
</tr>
<tr>
<td>Eval-1</td>
<td>50</td>
<td>5</td>
</tr>
<tr>
<td>Eval-2</td>
<td>50</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 7: Evaluation of Key(s) Selection: Chap: Chapter, Eval: Evaluator, G and B are for good and bad key 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 key is not always correct. For example, in sentence 14 selection of 6 as the key generated trivial GFQ.

**4.3 Distractors Selection**

Our system generates four alternatives for each gap-fill 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 GFQ (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 gap-fill questions generated by our system are shown above (red colored alternatives are good distractors, blue colored ones are the correct answers for the questions and the black ones are bad distractors).

<table>
<thead>
<tr>
<th>Class</th>
<th>Chap-5</th>
<th>Chap-6</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Eval-1</td>
<td>21</td>
<td>19</td>
<td>12</td>
</tr>
<tr>
<td>Eval-2</td>
<td>20</td>
<td>19</td>
<td>13</td>
</tr>
</tbody>
</table>

Table 8: Evaluation of Distractor Selection (Before any corrections)

Table 8 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 gap-fill questions with 0.67 inter evaluator agreement. Useful gap-fill 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 key 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.

In our work, as we only considered syntactic and lexical features for distractor selection, the selected distractors could be semantically conflicting with themselves or with the key. For example, due to the lack of semantic features in our method a hypernym of the key 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.

5 Related work

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. (2010) and Pino et al. (2009) used gap-fill questions for vocabulary learning. Smith et al. (2010) present a system, TEDDCLOG, which automatically generates draft test items from a corpus. TEDDCLOG takes the key 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 (GFS) or in distractors) in the generated gap-fill questions.

Pino et al. (2009) describe a baseline technique to generate cloze questions (gap-fill 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 (GFS). The new strategy produced high quality cloze questions 66% of the time.

Karamanis et al. (2006) 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. (2006). Initially key set is enlarged with NPs featuring potential key terms as their heads and satisfying certain regular expressions. Then sentences having at least one key 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 GFQs 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.

6 Conclusions and Future Work

Our GFQG system, selects most informative sentences of the chapters and generates gap-fill questions on them. Syntactic features helped in quality of gap-fill 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.

Acknowledgements

We would like to thank Avinesh Polisetty and Sudheer Kolachina from LTRC, IIIT-Hyderabad for their helpful discussions and pointers during the course of this work. Thanks also to the anonymous reviewers for useful feedback.

References


Lin, Y. C., Sung, L. C., Chen and M. C. 2007. An

Juan Pino, Michael Heilman and Maxine Eskenazi. 2009. A Selection Strategy to Improve Cloze Question Quality. Wkshop on Intelligent Tutoring Systems for Ill-Defined Domains. 9th Int. Conf. on ITS.


Ruslan Mitkov, Le An Ha and Nikiforos Karamanis. 2006 A computer-aided environment for generating multiple-choice test items, Natural Language Engineering 12(2): 177-194


Lee Raymond Dice. 1945. Measures of the Amount of Ecologic Association Between Species