Text2Test: Question Generator Utilizing Information Abstraction Techniques and Question Generation Methods for Narrative and Declarative Text

Jessica Franz Aquino
Information and Computer Studies, Faculty of Engineering, University of Santo Tomas
+639159646678
jessicafranzaquino@yahoo.com

Darwin Dwayne Chua
Information and Computer Studies, Faculty of Engineering, University of Santo Tomas
+639151297182
darwindwayne@gmail.com

Richard Kevin Kabiling
Information and Computer Studies, Faculty of Engineering, University of Santo Tomas
+639053410650
richardkabiling@gmail.com

John Nikole Pingco
Information and Computer Studies, Faculty of Engineering, University of Santo Tomas
+639158045121
johnnikolepingco@gmail.com

Ria Sagum
Information and Computer Studies, Faculty of Engineering, University of Santo Tomas
+639228181342
riasagum31@yahoo.com

ABSTRACT
In this study, the proponents developed a system that generates questions from declarative or narrative texts. The system implements information abstraction techniques such as anaphora resolution and factual statement extraction for the processing of its data. It also uses question generation methods gathered throughout the study. The system parses a text into its corresponding parse tree with the help of the Stanford Statistical Parser and abstracts and scores it accordingly. All sentences from the text with a non-complying score are removed. From what remains, as much questions as possible are generated. The study was evaluated by college professors and university students to determine the accuracy and difficulty of the generated questions.

Categories and Subject Descriptors
B.2.4 [**High-Speed Arithmetic**]: Algorithms
I.2.7 [**Computing Methodologies**]: Natural Language Processing

General Terms
Algorithms, Theory

Keywords
Question Generation, Text Processing, Scoring, Overgeneration

1. INTRODUCTION
Over the years, there are only several studies made concerning automatic question generation (eg, Mitkov & Ha [11]; Kunichika, Katayama, Hirashima, Takeuchi [8]; Rus and Greassar [12]; Rus and Lester [13]) and thus the realm QG in NLP is still young and not yet fully explored.

According to Heilman and Smith [6], the formulation of interrogative sentences (questions) has long been a major topic in the study of languages.

Question generation involves the purposive production of questions from a given input. In this research, a 3-step process was developed for a system (Text2Test) that takes an input narrative English text of variable length and from it produce questions based on a given importance constraint. This method is a rough replication of Michael Heilman and Noah Smith‟s [6] attempt at question generation (Overgenerate and Rank) with the application of several abstractive methodologies such that in NewsBytes‟ abstraction [2].

Assessments are vital in the learning and education process because not only these assist in measuring their students‟ performance, but also helps to understand the different factors including problems in those said processes [3]. The generation of question is a task that has a very significant role on several other fields of interest especially in education by providing materials that allow practice and assessment especially including that of literacy instruction. Although tests are a narrower concept than assessments, both are systematic procedures that aim to observe, describe, measure and obtain information for making (better) decisions on students in the field of education through more tangible schemes [12].

The final design is consist of three main parts, the first one is the Text Processing part which is based on the Factual Statement Extractor [7] and Anaphora Resolution (ARKref) [5] of Heilman and Smith, the second one is the scoring part where sentences are scored according to their weight on the text, and the third one is

Proceedings of the 8th National Natural Language Processing Research Symposium, pages 29-34
De La Salle University, Manila, 24-25 November 2011

the question generation proper which is heavily based on Overgenerate and Rank [6].

2. QUESTION GENERATION

Question generation, in Text2Test, is the process that encapsulates text processing, scoring and question overgeneration that encompasses this system‟s task of automatically generating questions from a set of input, see Figure 1.

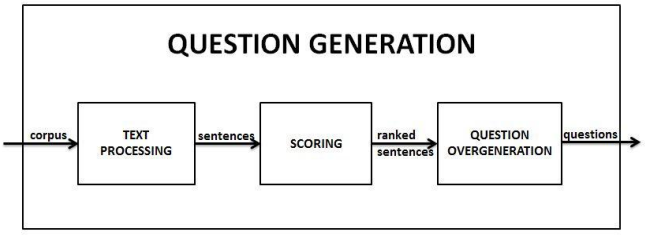

Figure 1. Question Generation System Flow

2.1 Text Processing

Text processing takes care of all the processing required and recommended before the overgeneration process. This process includes parsing, splitting and resolution of a text that the module takes as input (see Figure 2).

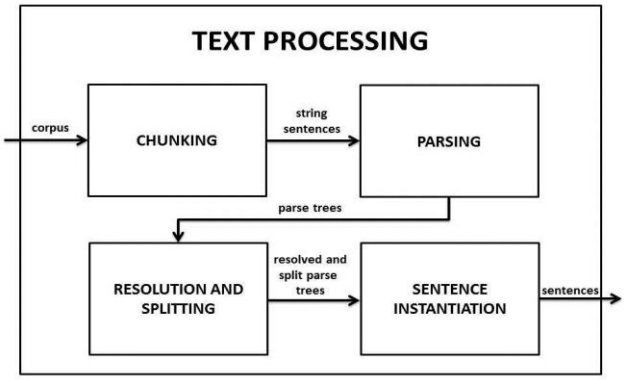

Figure 2. Text Processing System Flow Chart

2.1.1 Chunking

Chunking splits the input corpus into its constituent atomic sentences using regular expressions to enforce sentence boundary disambiguation rules. More complex rules involved arise from the ambiguous period which also marks abbreviations and ellipses.

2.1.2 Parsing

Parsing uses the Stanford Statistical Parser [8] to transform the chunked strings into their corresponding parse trees.

The parse tree is labeled with the set of standard Penn Treebank tags. These labels are further used in later phases of the question generation process such as in question overgeneration however, not all labels are relevant to the question generation process.

2.1.3 Splitting and Resolution

Splitting and resolution is concerned with decomposing complex sentences into much simpler sentences and resolving pronoun anaphora found within the text. This module is not entirely necessary in the question generation process. However, it does help in the abstraction of the whole text by clearing things up through the derivation of the text‟s atomic parts as well as removal of unneeded anaphora. This also allows more accurate scoring. The Splitter uses a system called the Factual Statement Extractor developed by Heilman & Smith [7] to split sentences and resolve anaphora.

2.1.4 Sense Tagging

During the identification of each token constituting a sentence, one of the properties of a token needed is its sense. The sense of a token is either its named entity tag or its supersense tag, whichever is detected by the system, with priority to the former. The system uses the Stanford Named Entity Recognizer [4] and the SuperSense Tagger from ARKref [5] to determine a word‟s token.

The Stanford Named Entity Recognizer, being a Named Entity Recognizer is only able to recognize proper nouns or words that are “named.” The SuperSense Tagger fills in this limitation by being able to tag a word of its sense even if it is not a proper noun. The SuperSense Tagger also tries to tag adjectives, adverbs and verbs, with more detail, by marking the main tagged word and the succeeding tagged word in the case of multiple words within the same sense. The Stanford Named Entity Recognizer model the system implements has seven tags while the SuperSense Tagger has 45 tags [5].

2.1.5 Morphology

For the lemmatization of certain words, as needed later on in the overgeneration process, Morphology, as included in the Stanford Statistical Parser, is used.

2.2 Scoring

The scoring process, see Figure 3, handles the scoring of all sentences. It is based on the premise of abstractive summarization systems that remove sentences that are deemed unimportant, only in the case of the Scorer, it removes sentences that do not fall within the specified range.

Scoring requires that the sentences are scored first. The score of a sentence is the average of the scores of its tokens, and the score of each token is based on how much it persists within the texts, particularly its normalized term frequency.

The sentences are then sorted based on these scores. The percentile rank of each sentence based on its score is afterwards retrieved (thus the sorting). From there, sentences with too low or too high ranks are removed from the bunch.

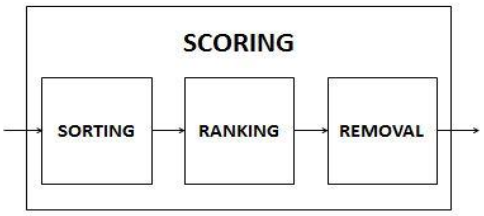

Figure 3. Scoring System Flow Chart

The score of each sentence is computed via the average of its constituting words, while the score of each word is based on the term frequency.

2.3 Overgeneration

After the scoring of the sentences, each sentence is prepared then continuously cloned and transformed or manipulated to form questions. To transform the sentences, particularly, their corresponding parse trees, Tregex and Tsurgeon [10] are used, the former being a tree querying language, and the latter being a tree manipulation language built around Tregex. This process is

heavily based on Heilman and Smith"s Overgenerate and Rank [6] however several modifications and improvements are added whenever the necessity arises.

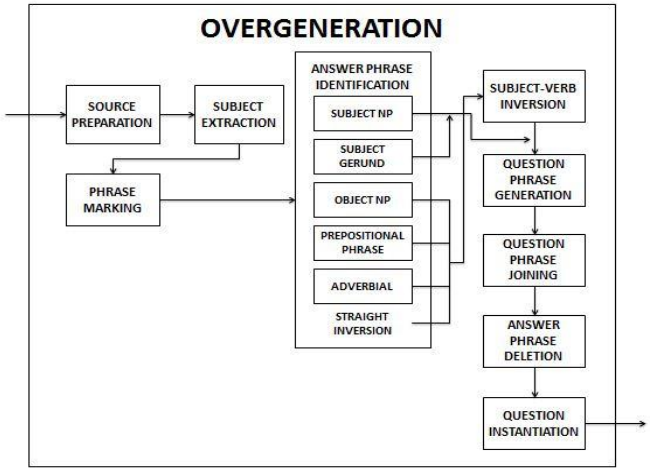

Figure 4. Overgeneration System Flowchart

Overgeneration has eight main subprocesses, see Figure 4. It is important to note that one of these phases, the answer phrase identification phase has nine variations, and subject-verb inversion is not compulsory for all cases. At the middle of the overgeneration process, the process tends to split to be able to appropriately generate several kinds of questions.

2.3.1 Source Preparation

Source preparation is a preprocessing phase before the main transformation phases begin. During the source preparation phase, certain parts of the sentence are omitted, or replaced so that processing later on will be easier due to the reduction of special cases and omission of unneeded parts. Several of the things done include: 1) replacement of contractions with their longer forms („re to are, „s to is), 2) removal of punctuation marks, 3) changing of first character to small if first word is not proper.

2.3.2 Subject Extraction

The subject extraction phase determines the subject, particularly the subject itself minus all its modifiers from the source sentence and identifies its multiplicity.

2.3.3 Phrase Marking

Phrase marking simply marks all parts of the parse tree that should be unmovable. This secures that all phrases under other phrases (ideally) should not move alone or be left behind when a movement happens. This also asserts that all other phrases within otherwise movable phrases (that are now marked us unmovable) are also unmovable. This is needed to avoid awkward questions such as in the question "What is St. Francis a peacemaker?" generated from "St. Francis of Assisi is a peacemaker."

2.3.4 Answer Phrase Identification

The answer phrase identification phase determines each part of the sentence that is viable to be an answer phrase then marks it accordingly on a clone of the sentence"s parse tree. Each produced parse tree from this phase corresponds to a unique answer phrase that will be used to come up with a question. This process can identify up to six types of answer phrases, but is not necessarily limited to producing only a maximum of six questions. Particularly, answer phrases are identified from noun phrases, gerund forms, and adverbial phrases. The different variations of the answer phrase identification phase corresponds to a certain type of phrase located on a particular position in a sentence and results to several questions of various kinds being produced per sentence (particularly Subject NP identifies NPs in the subject, Subject Gerund identifies gerunds in the subject, Object NP identifies NPs and gerunds in the predicate, Prepositional Phrase identifies PPs in the predicate and Adverbial identifies adverbials in the predicate; see Table 1). The answer phrase identification phase also clones the sentence an extra time, to be passed directly to subject-verb inversion to produce yes-no questions.

Table 1. Example of Answer Phrase Identification

Answer Phrase Identification	Source Sentence	Question Generated
Subject NP	***A space*** *conceals the jumping blast.*	What conceals the jumping blast?
Subject Gerund	***Having taken a Physics class*** *helped me in Calculus.*	What helped me in calculus?
Object NP	*The customers loved* ***the company's products.***	What did the customers love?
Prepositional Phrase	*Beef Pepper Rice was served* ***in a sizzling plate.***	In what was beef Pepper Rice served?
Adverbial	*The work was done* ***properly.***	How was the work done?
Straight Inversion	***The sky is blue.***	Is the sky blue?

2.3.5 Subject-verb Inversion

Subject-verb inversion is the next phase of the Overgeneration process. In this phase, the main linking verb or main auxiliary verb, if exists, is transferred to the beginning of the sentence. In the case that both do not exist, do, does or did, whichever is appropriate is added to the beginning of the sentence and the main verb is transformed to its lemma. Sentences with subject based answer phrases do not undergo this sub process.

Table 2. Guidelines for Question Phrase Generation

Answer Phrase Identification	Sense Tag	Wh Question	Preceded by Preposition
Subject NP	Person	Who	No
	Etc.	What	No
Subject Gerund	Etc.	What	No
Object NP	Person	Whom	No
	Etc.	What	No
Prepositional Phrase	Person	Whom	Yes
	Location or Place	Where	No
	Time	When	No
	Etc.	What	Yes
Adverbial	Etc.	How	No

2.3.6 Generation of Question Phrase

During the generation of the question phrase, the answer phrase is analyzed to come up with the appropriate question phrase particularly by using its sense tag (NER or SuperSense Tag) and its position its position in the source sentence to identify the appropriate interrogative pronoun to be used. Nominal answer phrases (answer phrases identified in Subject NP, Subject Gerund, and Object NP) always produce what and who questions (whom

in the case of objects). Prepositional phrase answer phrases are able to produce what, whom, where and when questions, the first two requiring the main preposition of the phrase (at what, with whom, over what). (See Table 2 for guidelines for Question Phrase Generation.)

2.3.7 Question Phrase Joining

Question phrase joining attaches the generated question phrase to the beginning of the sentence. The question mark is also added at the end of the sentence.

2.3.8 Answer Phrase Deletion

The identified answer phrase is deleted from the tree.

The yield of the tree is taken to retrieve the string form of the question. Final minor modifications and cleaning are done beforehand and afterwards. The tokens retrieved from the tree are concatenated to finally form the question.

2.4 Summary of Improvements

The study was able to improve its bases by adding several modifications to the methodology, adding another focus and asserting several other rules. First, added in the methodology is the abstraction and weighting (through scoring) and thus the added focus to abstraction more particularly to item difficulty estimation. A sense tagger along with Named Entity Recognizer (instead of just a Named Entity Recognition System) was incorporated to further increase the understanding of each sentence and word.

Some of the rules that were added to the overgeneration method include the processing of gerunds, inclusion of more modals, more detailed question phrase-answer phrase correspondence (for example, before place always equates to where, now, if in subject, now what, if in preposition, where; before, what is just what but now if in preposition, preposition is included in answer phrase).

3. PRESENTATION AND ANALYSIS OF DATA

3.1 Evaluation Methodology

Evaluators are given five sets of questionnaires together with the narrative text for each set of questions. The evaluation process was consisted of three main phases, the first phase deals with the evaluation of the accuracy of the generation process. The second phase concerns the evaluation of the item difficulty of each questions generated. Lastly, on the third phase, comments and qualitative evaluation and analysis of the faculties will be gathered.

3.1.1 Accuracy

The questions are evaluated on a 1 to 4 scale basis (4 being the highest) on syntactic and semantic correctness [1].

Syntactic correctness of each question was evaluated to ensure that Text2Test can generate grammatically correct output. The following scores was used to measure the syntactic correctness of each question: 4 – grammatically correct and idiomatic/natural, 3 – grammatically correct, 2 – some grammar problems, 1 – grammatically unacceptable [1].

On the other hand, semantic correctness of each question was also evaluated to make sure that each question has the correct meaning. The semantic correctness was evaluated using the following scores: 4 – semantically correct and idiomatic/natural, 3 – semantically correct, 2 – some semantic problems, 1 – semantically unacceptable [1]. The mean of the results of semantic and syntactic correctness were added to get the overall rating of Text2Test"s accuracy.

3.1.2 Item Difficulty

To make sure that the questions generated by Text2Test are answerable, the item difficulty of each question was evaluated. In evaluating the item difficulty of each question the following scores was used: 4 – very difficult, 3 – difficult, 2 – easy, 1 – very easy.

3.1.3 Reinforcement

Furthermore, like Mitkov and Ha [10] did in measuring the accuracy of their QG, the proponents will be asking faculty evaluators for commentary and qualitative evaluation and analysis over the questions produced by Text2Test.

3.2 Results of the Conducted Research

Text2Test was evaluated in terms of the accuracy and the difficulty of questions produced. The evaluation was performed with faculties and students which has enough knowledge and understanding in English language.

3.2.1 Accuracy

The results of the evaluation among students showed an accuracy of 6.4626 over a total of 8 or an accuracy of 80.7825%. This accuracy was derived from the average syntactic correctness of 3.2391 out of 4 and syntactic correctness of 3.2235 out of 4, translating to 80.9775% and 80.5875% respectively. Out of the 25 responses composing of 5 texts resulting to more than 300 questions per response from students, 13 or 52% of the time, syntactic accuracy is greater than the semantic correctness. The difference between syntactic and semantic correctness showed a difference of only 0.0156 or 0.39%.

Table 3. Accuracy Evaluation Results

	AVERAGE SEMANTIC	AVERAGE SYNTACTIC	TOTAL ACCURACY
Faculties	2.9138	2.8725	5.7858
Students	3.2235	3.2391	6.4626
AVERAGE	3.0687	3.0558	6.1242

On the other hand, the results from faculty members showed lower accuracy than that of students. The accuracy as evaluated by teachers showed that out of 8, the questions on average only score a 5.7858. This translates only to 72.3227%. The syntactic correctness suffered more from their evaluation, only getting 2.8725 out of 4 or 71.8131% contrary to semantic correctness which gained a little more, 2.9138 out of 4 or 72.846%. Out of the 15 responses composing of 5 texts resulting to more than 300 questions per response from teachers, 7 out of 15 or 46.6667% of the time, semantic accuracy is greater than the syntactic correctness. The difference between syntactic and semantic correctness showed only a difference of 0.0413 or 1.0329%.

The shortcomings of the system in terms of syntactic accuracy could probably be attributed to lack of verb transformation feature in case of changes in multiplicity resulting to sentences with incorrect subject-verb agreement. Awkward sentence syntax could also result to incorrect simplification of complex sentences. Lemmatization, incorrect parsing, incorrect tagging and similar cases (cases where in the modules could not correctly assert their functions due to the uniqueness or ambiguity of the situation) could also be minor but still common factors.

The shortcomings of the system in terms of semantic accuracy could probably be attributed to insufficient sense recognition, complex sentence simplification or pronoun anaphora resolution capabilities as the inaccuracies of the tools used for the system are also unfortunately carried over into the system. Ambiguity could also be a problem such as in the case of complexly nested phrases that are difficult to trim down to its atomic noun phrase. Lack of rules to resolve such ambiguities may also be a cause for a lower semantic accuracy.

The lack of selection process after the generation process were also factors for both syntactic and semantic correctness as bad questions still make it as output due to not being selected out before the end of the whole process.

On average, the results showed a syntactic correctness of 3.0558 out of 4 or 76.39% and a semantic correctness of 3.0687 out of 4 or 76.7168%, which gives a total accuracy of 6.1242 out of 8 or 76.5526% (see Table 3). These results showed a good degree of accuracy and fairs well with respect to other question generation systems.

The results showed that the system performs slightly better in terms of semantic correctness than syntactic correctness. The former leads with only 0.0129 out of 4 or 0.3214%. This asserts that the system was able to assert a balance between syntactic and semantic correctness so that deriving questions have good syntax and semantics.

The lack of precision in the evaluation choices could also have potentially decreased the recorded accuracy as perceived by the evaluators.

3.2.2 Item Difficulty

On average, faculty members and students rated the 1st scale 2.37 (1 as easiest, 4 as hardest), 2nd scale 2.37, 3rd scale 2.25, 4th scale 2.34, 5th scale 2.02, 6th scale 2.50, 7th scale 1.88, 8th scale 2.42, 9th scale 1.93 and the 10th and final scale 2.40(See Figure 5). Eight out of 10 of these or 80% ranked between easy and difficult, while the other 2 or 20% ranked between very easy and easy. When averaged, the questions scored with 2.50 which falls between easy and difficult (moderate).

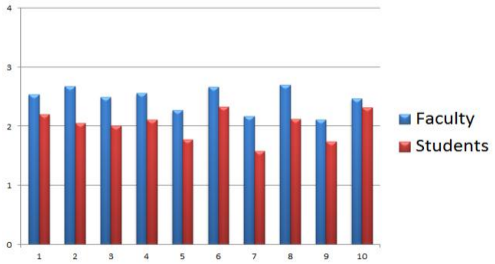

Figure 5. Item Difficulty Result

The difficulty of the questions didn‟t come out as expected. Supposedly, questions with more frequent words thus higher scores are easier than the opposite, as asserted in other researches; however the gathered data showed arbitrary difficulty as perceived by students and the faculty. There are slight manifestations of this; however they are not too continuous and inconsistent to even be considered as plausible support.

3.2.3 Reinforcement

After their evaluation of the questions generated using the previously mentioned metrics, the teachers were asked to give reinforcement to their evaluations and share their observations and insights regarding the system. Their comments are valuable because they are considered field experts and that they as teachers, will be the main users of the system.

The teachers shared and described their observations on the two things being evaluated, accuracy (through semantic and syntactic correctness) and item difficulty.

According to one, the questions produced in terms of semantic correctness were good. All questions were really based on the text and all answers can somewhat be found in the text. However, one commented that the questions were quite similar. This could probably attributed to the lack of selection process after the overgeneration stage of the question generation process.

Another contributed that semantically, a big percentage of the questions were semantically correct but there were quite a number of semantically incorrect questions. Some questions tend to be vague as well due to lack of modifiers. The faculty member specifically said that the system should focus on producing more semantically correct questions.

A common semantic accuracy related problem is incorrect WH-questions. One of the teachers even asserted that there is a blatant misuse of the Who and What.

The blunt relationship between the answer and the text also gave them the impression that there is no need for explanation. They also found that the questions were overwhelmingly text based as they do not require going deeper into the text because most of the questions are taken from the text verbatim. There are also instances where the questions are irrelevant to what the text is talking about despite still being perfectly based off the text such as in the case of adverbials as answers. One of the teachers perceived that all the questions on average could be rated with 3 out of 4.

The syntactic correctness according to the teachers was good and acceptable most of the time. There were occasional grammatical mistakes that could be serious such as when long sentences from the text are split and the product sentences were either wrong or confusing. There are also common minor mistakes that do not obscure understanding at all such as in the case of S-V agreement problems. In a few cases, syntactic arrangement causes confusion. However, despite these shortcomings, the syntactic correctness is still very acceptable. There also seems to be a problem with some hyphenated words coming out wrong after the generation process. This could be because of some problems with the lemmatizer. A faculty member asserted that many of the syntactic problems were really tightly to related to semantic problems as well (such as WH-questions) and could domino to item difficulty, because difficult to understand questions become hard to answer questions as well. One rated the correctness of all the questions in general with 3 out of 4.

With regards to the difficulty of the questions generated, since the questions are just generated through the inputted text, the odds are if the student or whoever is answering the questions generated by system has fully read and understood the text, he or she will find answering the questions easy. Some may be a little bit of a tickle to the brain but on the average, the questions are easy. Just a little observation: some of the questions though may be structurally and grammatically distinct from each other, have the same answers. They have also observed that by and large, the questions are answerable by yes or no, by the subject of the sentence or the direct object. Another observation is some of the questions tend to

be wordy and contains too much information on them that the succeeding questions might draw answers from them.

They also suggested that the questions were quite repetitive and too interconnected. Some even had the same answers causing problems.

The teachers affirmed several things about the system. They added that the system is good, and can be used in education especially students whenever they need guide questions. The questions were good enough and just require another layer such as physical selection and editing to further refine the questions. One of them stated that the system can be useful when constructing question banks. It could also be useful for testing students" reading or listening comprehension skills.

Generally, they said that the system, its capabilities and the questions it produced were sufficient and efficient thus acceptable.

4. CONCLUSIONS AND FUTURE WORKS

4.1 Conclusions

Text2Test generated questions which on average had an accuracy of 3.0558 out of 4 or 76.39% and a wildly and inconsistently varying difficulty which averaged to 2.5 (between easy and difficult) contrary to expected which is difficulty inversely varying to the score (more difficult, lower score).

Text2Test was also able to give balanced questions when it came to semantic and syntactic accuracy as the difference between the two factors of accuracy is barely noticeable.

4.2 Future Works

The results of the study allowed the discovery of several more fields for further study. To produce more accurate questions, grammar correction beforehand for the text is advised as well as processing of quotations, special and stylized characters. Implementation of other abstraction and scoring techniques with other question generation methods may also give better results. The system could further be improved by incorporating more accurate pronoun anaphora resolution and resolution for logical assertions. Furthermore, a more intelligent factual statement extractor and chunking methodology could be used to extract simple sentences from complex sentences and sentences from paragraphs. In addition other welcome features include a verb transformation feature that can help on having a more accurate subject-verb agreement, a better factored parser, and a better lemmatization module. Developing, improving and implementing better question generation methods like finding other answer phrases like verb-based answer phrases can be used to generate other types of questions. Adding new stages like feature based ranking can improve and include final selection. Other than the above mentioned, a better scoring system that also takes the answers into consideration when scoring may help in furthering the study"s abstraction on the difficulty of an item. The system could also be redone for other languages.

5. ACKNOWLEDGEMENT

The researchers would like to thank the following: Mr. Cecil Jose Delfinado, Ms. Charmaine Ponay, Ms. Jerralyn Padua and Ms. Donna Acula for their suggestions, comments and input regarding the process involved – we thank them for their countenance, The Residences @ P. Campa, Miranda and Chua residences and the University of the Santo Tomas for providing for us an environment where we could conduct our research – we thank them for their hospitality, our family and friends for being there to support us – we thank them for their presence and finally God for giving us strength, bringing us guidance and showing us compassion to get through this.

6. REFERENCES

[1] Agarwal, M., Shah, R., & Mannem, P. (2011). *Proceedings of the Sixth Workshop on Innovative Use of NLP for Building Educational Applications: Automatic Question Generation using Discourse Cues.* Retrieved from http://www.aclweb.org/anthology/W11-1401

[2] Batang, E., Bonus, D.E., Cruz, R., Miano, M. A., Sagum, R., & Yu, R. (2007).Proceeding of the forth Natural Language Processing (NLP) Research Symposium: *NewsBytes: Tagalog Text Summarization Using Abstraction.* Retrieved from http://www.dlsu.edu.ph/

[3] Cheng, S., Lin, Y., & Huang, Y. (2009). Dynamic question generation system for web-based testing using particle swarm optimization. *Expert Systems with Applications*, 36(1), 616-624. doi:10.1016/j.eswa.2007.09.064

[4] Finkel, J.R., Grenager, T., & Manning, C. (2005). Stanford Named Entity Recognizer (NER) [Software]. Available from http://nlp.stanford.edu/software/CRF-NER.shtml

[5] Heilman, M., & O"Connor, B. (2009). ARKref [Software]. Available from http://www.ark.cs.cmu.edu/ARKref/

[6] Heilman, M. & Smith, N.A. (2009). *Question generation via overgeneration transformation and ranking* [PDF Document]. Retrieved from http://www.cs.cmu.edu/~mheilman/

[7] Heilman, M. & Smith, N.A. (2010). Factual Statement Extractor [Software]. Available from http://www.ark.cs.cmu.edu/mheilman/qg-2010-workshop/

[8] Klein, D., & Manning, C. (2003). The Stanford Parser: A statistical parser [Software]. Available from http://nlp.stanford.edu/software/lex-parser.shtml

[9] Kunichika, H., Katayama, T., Hirashima, T., & Takeuchi, A. (2004). Proceeding of ICCE: *Automated question generation methods for intelligent English learning systems and its evaluation*. Retrieved from http://www.minnie.ai.kyutech.ac.jp

[10] Levy, R. & Andrew, G. (2006). Tregex and Tsurgeon [Software] Available from http://nlp.stanford.edu/software/tregex.shtml

[11] Mitkov, R. & Ha, L.A. (2003). *Computer-aided generation of multiple-choice tests.* HLT-NAACL-EDUC '03 Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing, 2(1), 17-22. doi:10.3115/1118894.1118897

[12] Rus, V. & Graesser, A.C. (Eds.) (2009) *The question generation shared task and evaluation challenge*. Retrieved from http://www.questiongeneration.org

[13] Rus, V. & Lester, J. (Eds.) Proceedings of the 2nd Workshop on Question Generation In Craig, S.D. & Dicheva, S. (Eds.) (2009) AIED 2009: *14th international conference on artificial intelligence in education: Workshops proceedings*. Retrieved from http://www.questiongeneration.org