|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Title** | | **Problems** | | **Solutions** | | **Recommendations** | |
| Automatic generation of multiple-choice tests  2010  By: S´ergio dos Santos Lopes Curto | | * Developing a computer-aided multiple choice exam generator * to create a platform to assist on multiple choice test items generation based on an input corpus. | | * Curto divided the process into two main parts: Question Generation and Distractor Selection * Question Generation process is divided into several modules. This includes Information Extraction to configure a “situation model”, a Question Design module that uses Natural Language Generation techniques, and an Evaluation Module that filters low quality questions automatically or manually by the user. * Distractor Selection process consists in locating in the corpus the words that share traces with the answer to the question. This simple method allows the generation of distractor candidates. | | * Create an user-friendly interface for the rules based system * Study the priority of the features of the answer that should be present on the distractors, increasing the number of distractors generated while keeping the quality of the generation; * Application of an anaphoric reference solver component to improve the quality of the test items generated; * Explore additional distractor sources (for example WordNet); | |
| A real-time multiple-choice question generation for language testing  a preliminary study  by: Ayako Hoshino,Hiroshi Nakagawa | | * Automated Multiple Choice Question Generation in testing a student’s proficiency in a foreign language * extract important words or phrases in a text for a learner of the language. * To generate both multiple choice and cloze questions. | | * The process of converting the input to multiple-choice questions includes extracting features, deciding the blank positions, and choosing the wrong alternatives (which are called *distractors*). * Hoshino et. al. proposed using a standard Machine Learning framework that involves training. * For cloze questions, The positions of the blanks are decided according to the certainty of the classification so the blanks (i.e. questions) are generated as many as the user has specified. * the distractors are chosen randomly from the same article excluding punctuations and the same word as the other alternatives. * To examine the quality of the questions generated by the current system, the researchers have evaluated the blank positions determined by a Naive Bayes classifier and a KNN classifier (K=3) with a certainty of more than 50 percent in 10 articles. | | * In the current system, the mechanism of choosing distractors is implemented with the simplest algorithm, and its investigation is left to future work. * use larger number of features, possibly including semantic ones, so a blank position would not depend on its superficial aspects. * A semantic *distance* between an alternative(distractor) and the right answer are suggested (Mitkov and Ha, 2003), to be   a good measure to evaluate an alternative. | |
| Automatic Gap-fill Question Generation from Text Books   * Manish Agarwal and Prashanth Mannem | | * 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. | | * 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* by going through all the sentences in the documents and extracts a set of features from each of them. * The approach in *key selection* involves two processes: In the first step the module generates a list of *potential keys* from the gap-fill sentence (*key-list*) and in the second step it selects the best *key* from this *key-list*. * For *distractor selection,* three comparisons for the key and a possible dsitractor are used: Contextual Similarity, Sentence Similarity and difference between term frequencies | | * Further experimention on larger data by combining the chapters of a text book. * Evaluation of course coverage by the system and use of * semantic features. | |
| Question generation via overgeneration transformations and ranking  -M. Heilman & J. Smith(2009) | | Generate questions for the purpose of  reading comprehension assessment and practice. | | Heilman and Smith used modular, three-stage  framework for automatic comprehension question  generation: (1) extract and derive declarative sentences from a source text; (2) transducer declarative sentences into questions using declarative, general-purpose rules; and (3) statistically rank the output of overgenerating stages 1 and 2 for acceptability. | | For future work, they recommend to extend the unit tests by examining  the data from the evaluation so it can safely modify and extend the rule set to improve coverage and accuracy. | |
| Automatic Factual Question Generation from Text –M.Heilman(2011) | | Create a system for question generation that can take as input an article of text  (e.g., a web page or encyclopedia article that a teacher might select to supplement the materials in a  textbook), and create as output a ranked list of factual questions. | | Heilman used 3 stages in his system. The first stage is to transform set of sentences into simpler declarative statement. The implementation includes operations for extraction and simplifying complex sentences and for resolving pronouns. The declarative sentence is turn into a set of questions by executing a series of well-defined syntactic transformations (WH-movement, subject-auxiliary inversion, etc.). Last process involved in Heilman’s system is to score and rank questions according to features of the source sentences, input sentences, question, and transformations used in generation | | Heilman recommended having an alternative representation for QG, information extraction for QG and other NLP transformations such as sentence compression, sentence fusion, or paraphrase generation in transforming set of sentences into simpler declarative statement. | |
| A New Approach to Ranking Over-Generated Questions  McConnell, Mannem, Prasad, Joshi, (2011) | | * Some questions generated in existing systems are unclear and does not make any sense. * Grammatically error * Wrong WH-words (What and When) | | * Sentence scoring by their topic score * A bigram model to calculate question phrase probability and uses MS Encarta question database to form a question * implemented search key function (eg. dates, time, etc) and WordNet Hierarchies in What questions | | * Implement a procedure for evaluating the language model’s accuracy in measuring grammaticality. * Incorporate Wordnet hierarchies to improve *Who* questions and explore other potential applications of this database within QG. | |
| Good Question! Statistical Ranking for Question Generation  By Heilman and Smith | | Generate fact-based questions about the content of a given article. The top-ranked questions could be filtered and revised by educators, or given directly to students for practice. | | * QG as a two-step process of first simplifying declarative input sentences and then transforming them into questions, the latter step being achieved by a sequence of general rules. * Apply statistical ranking to the task of generating natural language questions. | |  | |
| Corpus-Based Identification of Non-Anaphoric Noun Phrases  By: David L Bean and Ellen Riloff x(2002) | | Existence of Non-Anaphoric Noun Phrases.  Most automated approach to Coreference resolution attempt to match a corresponding antecedent for every possible referent discourse entity. The problem with this approach is a large number of the said discourse entities refer to something that can only be understood from a world general knowledge. | | Identifying Existential Noun Phrases  The goal is to build a system that can identify independent existential noun phrases automatically. This is done with the use of four methods; (1)Syntactic Heuristics contains processes such as Restrictive premodification and postmodification.  (2)Sentence One Extractions, creates a list of presumable existential NP by collecting the first sentence of every text and extracting all definite NP not classified by the first method.  (3)Existential Head Patterns, this is done by building candidate head patterns for every NP of more than two words in this way, we recognize compound head nouns.  (4) Define-Only List, The first pass produced a list of every  definite NP and its frequency. The second  pass counted indefinite uses of all NPs cataloged  during the first pass. Knowing how often an NP  was used in definite and indefinite constructions  allowed us to sort the NPs, first by the probability  of being used as a definite (its definite probability),  and second by definite-use frequency. | |  | |