iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 Issue: V Month of publication: May 2015

DOI:

Volume 3 Issue VI, June 2015
ISSN: 2321-9653

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

A Heuristic Approach to Factoid Question Generation from Sentence

Aritra Das[#1], Animesh Shaw[#2], Shreeparna Sarkar[#3], Tamal Deb[#4]
[#]Computer Science & Engineering Department, Narula Institute of Technology

Abstract*— Question Generation (QG) and Question Answering (QA) are among the many challenges in natural language generation and natural language understanding. An automated QG system focuses on generation of expressive and factoid questions which assist in meetings, customer helpline, specific domain services, and Educational Institutes etc. In this paper, the proposed system addresses the generation of factoid or wh-questions from sentences in a corpus consisting of factual, descriptive and unbiased details. We discuss our heuristic algorithm for sentence simplification or pre-processing and the knowledge base extracted from previous step is stored in a structured format which assists us in further processing. We further discuss the sentence semantic relation which enables us to construct questions following certain recognizable patterns among different sentence entities, following by the evaluation of question generated. We conclude our project discussing the applications, future scope and improvements.

Keywords*— Question Generation, Question Answering, Information Retrieval, Syntactic Parsing, Recursive Descent Parsing, Named Entity Recognition.

I. INTRODUCTION

Automatic generation of questions is a challenging and an important research area in natural language generation, potentially useful in intelligent tutoring systems, dialogue systems, educational technologies, instructional games etc. Generation of questions for different purposes from some large corpora by extracting useful or key information from them is a tiresome and challenging task. Manual methods may not always be applied on such occasions where questions from data are required in a short time but the advancements in the growth of computational power have paved the path to language learning and provide us an opportunity to automate such processes. We invested our interest in using question generation for tutoring and educational purposes primarily but not restricted to this domain.

A. Factoid or Wh-type Questions

Factoid questions are the most widely studied task in question answering. In this paper, we survey several different techniques to answer extraction for factoid question answering, which aims at accurately pin-pointing the exact answer in retrieved documents. These are questions that demand accurate information regarding an entity or an event, whose answers are of syntactic or semantic entities e.g. PERSON, DATE, TIME, LOCATION etc. as opposed to definition questions, opinion questions or complex questions as the why or how questions. Wh-questions differ depending on the kind of content information sought. Content information associated with persons, things, and facts is generally sought with one set of wh-words, and content information associated with time, place, reason, and manner is sought with another set of wh-words.

B. Types of Factoid or Wh-type Questions

1) Who type: Questions that are generally concerned with query about what or which PERSON?
2) What type: These types of questions seek for information about something. It generally seeks an answer that is generally in verb phrase, gerund, and infinitive. The answer related to these type of questions are usually restricted to one line but if "what" is combined with "for" or the question is generated with a query of definition, then the information that these questions are based on may range over major fraction of the given passage.
3) When type: These types of questions refer to nouns that are related to TIME. For e.g. time, date, season etc.
4) Where type: These types of questions are related to LOCATION entity.
5) Which type : These types of questions query about specification of a particular individual among a collection

In this paper, the proposed algorithm addresses the generation of factoid questions or Wh-Questions from sentences of a corpus or some data file given as input, which contains factual details. In order to facilitate the QG task, our algorithm takes a large corpus in raw or text format which is then further processed using a pre-processing algorithm which converts the raw text into a structured format we generate a syntactical parsetree for each sentence in a corpus using MBSP parser[13] which further applies text processing techniques like chunking, pos tagging and relation finding. We us this data produced and use a custom heuristic based question type identification system[3] to guess the question type for a sentence which creates a probability model framework, upon which the question structure are analysed and generated.

II. RELATED WORK

Question Generation remains a challenging area of research among the Artificial Intelligence, Natural Language Processing or Machine Learning. In a study by Smith, Heilman and Hwa [14] states question generation as a “Competitive Undergraduate Course Project” mainly because if inconsistency of the data and the different ways question generation can be approached. Chen, Wei, and Jack Mostow [2] used it to classify children’s spoken responses to a reading tutor teaching them to generate their own questions. Liu, Ming, Calvo, and Rus [1] created an automatic question generation system for students’ academic supports from papers and journals. They defined some rules and template based structure to extract certain types of data and generate questions from them. Agarwal, Manish, and Mannem [5] worked on an automatic question generation system that can generate gap-fill questions for content in a document. Syntactic and lexical features are used in this process without relying on any external resource apart from the information in the document. Xu, Yushi, Goldie, and Seneff [7] applied the concepts of question generation to create a web based game to learn Mandarin Chinese language. Lin, Chin-Yew [9] proposed an approach to automatic question generation from web search queries. Data along with search engine query logs to create a question generation shared task that aims to automatically generate questions given a query. Ali, Husam, Chali, and Hasan [10] proposed a method of question generation where a template based approach was taken to generate the questions from sentences. They used the phrase features and NER (Named Entity Recognition) of a sentence to identify the potential informative data and from them they generate the questions.

III.PROPOSED FACTOID QUESTION GENERATION MODEL

In our proposed approach we have laid stress on the syntactic features of sentences, rather than approaching with a grammar based question generation, we have worked on the syntactical parse tree that is created for each sentence. Grammar based approach could be more time consuming, but our algorithm works on the parse tree for each sentence. We feed into our system questions from the WebQuestions dataset [15] which creates a rule by itself used to generate the questions. Every path to the leaf from the root of the tree is a rule itself which is used for building the probabilistic model and generating the question. The complete working procedure and algorithm have been discussed as follows.

A. Generation of Grammar RuleBase Tree

In the first step we process the sentence we receive as input and perform some NLP techniques like Chunking, and Named-Entity-Recognition.

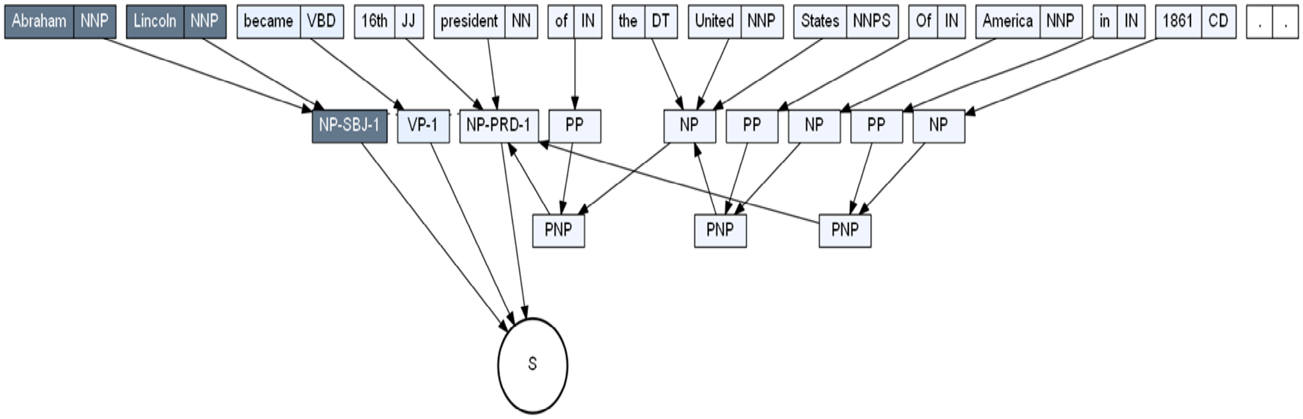


Fig. 1 Sentence Parse Tree

1) Here we took the relations like NP-PRD-1, VP-1, NP-SBJ-1 etc. to generate a tree which serves as the knowledge base and is stored in XML format, this knowledge base in further used as the rule base from which the system takes the help while generating the question. In the above image NP-PRD-1 is the relation name of the chunk ‘16th president’.

2) The Wh-type is determined by the observing the head word of the question. A list with chunk relations is made and saved into an ordered data structure. For the above example the list will be look like:
[‘NP-PRD-1’, ’VP-1’, ‘NP-PRD-1’, ‘PP’, ‘NP’]

3) That list and the question type or the Wh-tag is put into another list where the 1st item is the Wh-tag and the 2nd item is another list which contains the chunk relations. Now the list would look like:
[‘Who’, [’VP-1’, ‘NP-PRD-1’, ‘PP’, ‘NP’]]

4) A tree that is given to the system has root as Q and its children as Wh-tags.
[The pre generated empty tree with wh-tags]

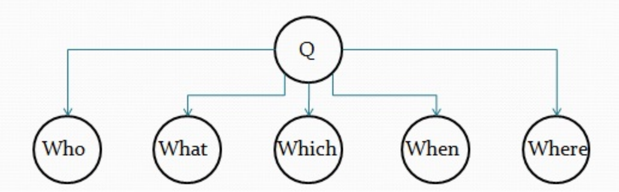


Fig. 2 Empty Tree with Question Tags

5) On seeing the Wh-tag from the list the systems goes to specific node and populates the tree with the chunk relations which is the 2nd item of the list. For each relation from the list a node is generated in the tree.

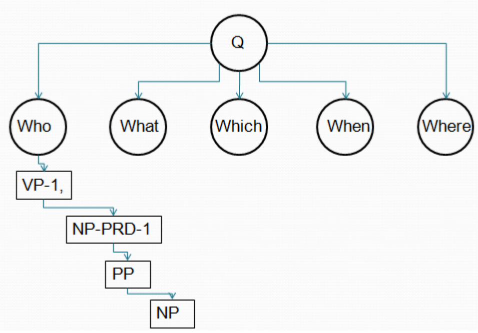


Fig. 3 Populating Chunk Relations

6) While populating, the count of visiting each tail node (the node occurs at the last of chunk relations) is saved in the corresponding node. A snapshot of the rule base with count value :-

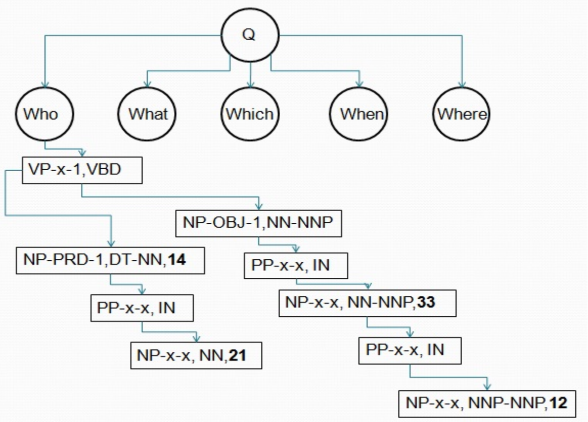
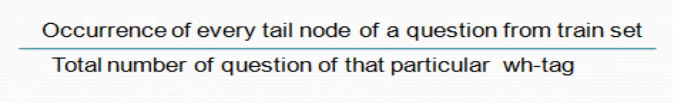


Fig. 4 Rule Tree with Visiting Node count value

7) Normalized count value is used to let the parser know when to print the question and when not while backtracking to other child nodes. It is defined as :



$$\frac{\text{Occurrence of every tail node of a question from train set}}{\text{Total number of question of that particular wh-tag}}$$

Fig. 5 Normalized count formula

Algorithm 1: Function to Generate tree from chunk relations**:**

```
/*Function takes a 'Sentence' as argument*/
TreeBaseGeneration (Sentence):
    Chunk list <= MBSP.parse(Sentence)
    Question type <= QTypeGen(Sentence)
    Chunk list <= [QuestionType, ChunkList]
    Main Root <= GetTreeRoot(QuestionTree)
    Wh-node <= MainRootschild(ChunkList[1])
    Root <= Wh-node
    For relations in ChunkList[2] :
        If relation is found:
            Root <= RootsChild (relation)
        Else:
            Child <= CreateChild(relation, Root)
            Root <= Child
    If the current Root is tail node:
        Increase Root's count by 1
    Return
```

B. Question Generation from a given Sentence

After the creation of rule base tree we move to the next step to generate the questions from a given sentence. The process is explained in the following steps:-

1) Answer Preprocessing and Question type decision system:

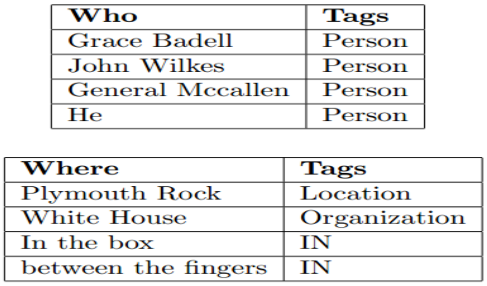
a) While populating the tree with manually generated questions, the NER tag of the answer for a given question is stored with the corresponding wh-tag. Here, the answer is not a complete sentence; it consists of only some word(s).
For example in the train set the questions and the answers are given as following:

"Who is the Father of the Nation? Ans Mahatma Gandhi."

From the above sentence 'Mahatma Gandhi' word is NER tagged:

Mahatma $_{\text{[PERSON]}}$ Gandhi $_{\text{[PERSON]}}$

b) If a same tag is found in the answer-r again and again, the count value is increased accordingly.



Who	Tags
Grace Badell	Person
John Wilkes	Person
General Mccallen	Person
He	Person

Where	Tags
Plymouth Rock	Location
White House	Organization
In the box	IN
between the fingers	IN

Fig. 6: Picture of the answer base.

c) The given informative sentence is parsed and divided into tokens. Then the stopwords are removed from the sentence.

d) The reduced sentence is fed to the MBSP parser and the chunk relations along with the corresponding chunks are extracted.

e) The probability of the type of the question is calculated through the following function :-

$$Probability\left(Word_1, Word_2, \ldots . \frac{S}{Wh_{Tag}}\right) = \frac{Count(Wh_{Tag})}{Count(All_{Tag})} \prod_{Word_i \in Words} \frac{Count(Tag(Word_i)) + 1}{Count(All_{TagTypes}) + |VocabularySize|}$$

Fig. 7. Probability Function

f) The Wh-tag and corresponding probability is stored in a table where keys are wh-tag and the values are probabilities.

g) Tag with maximum probability is taken into concern and that type of question is generated.

2) *Question generation:*

a) The question is generated from the rule base tree after getting the type of question is needed to be generated from the previous step. User input a sentence to the system and that sentence is chunked and the chunk relations are taken out.

b) The chunks and the corresponding relations are put into a table where the keys are the relations and the values are the chunk phrases.

c) The type of Wh-tag is selected and the tree is traversed by a recursive decent parser.

d) When a tail node is traversed the system prints out the question.

Chunks	**P-S-R**	**POS tag**
Who	NP-SBJ-1	WP
is known	VP-1	VBZ,VBN
as	PP-x-1	IN
the Father	NP-x-1	DT,NNP
of	PP	IN
Nation	PP	NN

Fig. 8 Table showing chunks

Algorithm 2: Function to generate the question string:

```
RecursiveDecentParser(Node, ChunkTable, QuesString):
    1.  If Node has no Child and Node is TailNode:
            Print QuesString
    2.  If Node has more than one Child:
            For each Child of Node:
                If ChunkTable has the Child :
                        RecursiveDecentParser(Child, ChunkTable, QuesString + ChildName)
                ElseIf Child is not in ChunkTable:
                        Print QuesSting
                Else:
                        If Node is TailNode:
                            Print QuesString
                            RecursiveDecentParser(Child, ChunkTable, QuesString + ChildName)
    3.  If the question is not printed already while traversing the Node:
            Print the question only if the current Node is a TailNode.
```

Working flowchart of the above Algorithm 2 to generate Questions

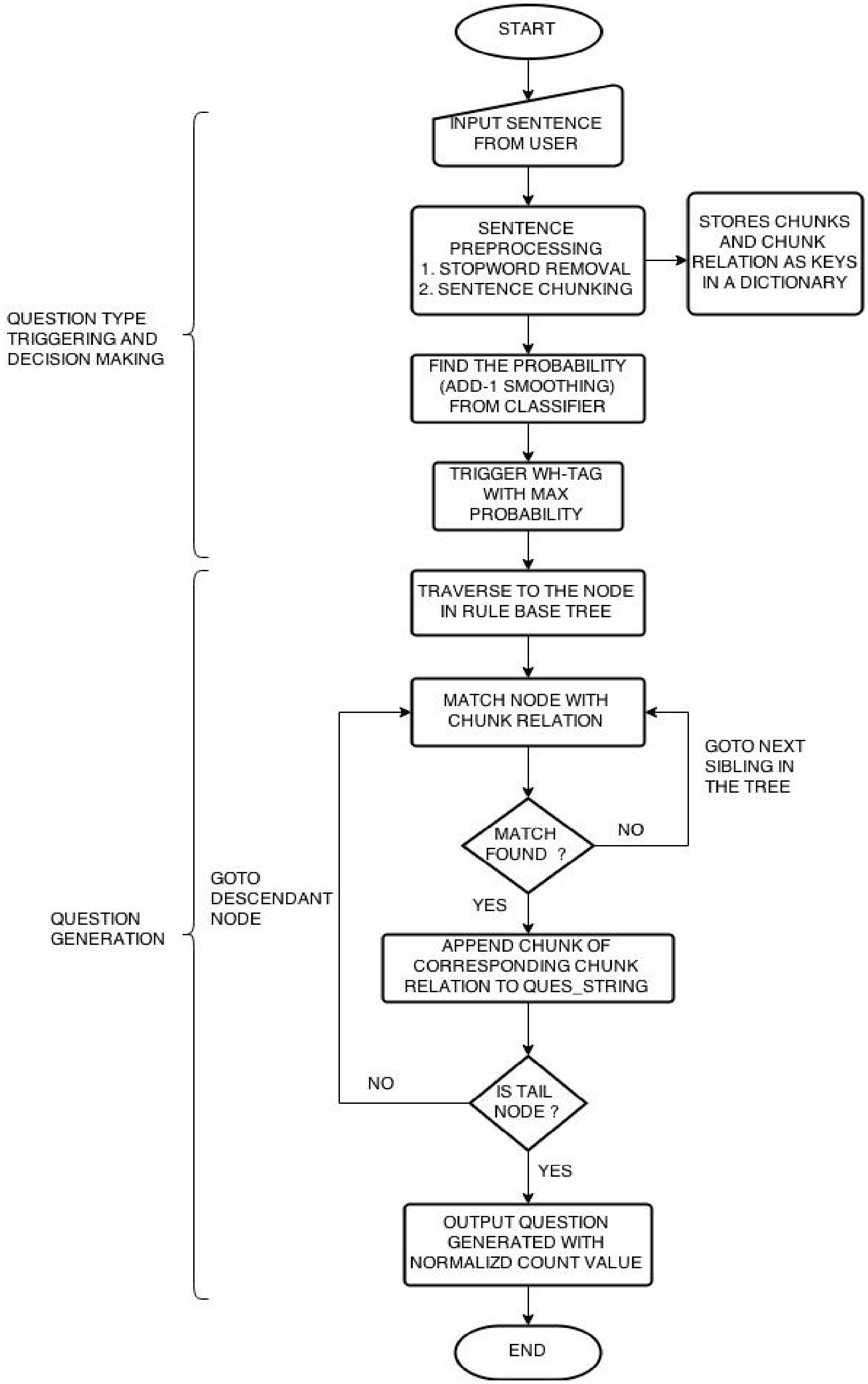

START
INPUT SENTENCE FROM USER
SENTENCE PREPROCESSING
1. STOPWORD REMOVAL
2. SENTENCE CHUNKING
STORES CHUNKS AND CHUNK RELATION AS KEYS IN A DICTIONARY
FIND THE PROBABILITY (ADD-1 SMOOTHING) FROM CLASSIFIER
TRIGGER WH-TAG WITH MAX PROBABILITY
QUESTION TYPE TRIGGERING AND DECISION MAKING
TRAVERSE TO THE NODE IN RULE BASE TREE
MATCH NODE WITH CHUNK RELATION
GOTO NEXT SIBLING IN THE TREE
MATCH FOUND ?
NO
YES
APPEND CHUNK OF CORRESPONDING CHUNK RELATION TO QUES_STRING
GOTO DESCENDANT NODE
IS TAIL NODE ?
NO
YES
OUTPUT QUESTION GENERATED WITH NORMALIZD COUNT VALUE
QUESTION GENERATION
END

Fig. 9 Flowchart to Generate Questions

IV. RESULT EVALUATION

We tested our algorithm against a sentence dataset manually generated. The following results were obtained when tested.

	Manual Generation	System Evaluation
Precision	100	57
Recall	100	88.46

V. CONCLUSION

In this paper, we propose a heuristic based system of generating factoid based questions from sentences. We trained our heuristic based system on a small dataset and the results were promising. With large data set our system would perform much better. Our approach can further be used in other question generation techniques like a template based approach, or can be extended to be used in Intelligent Systems.

VI. ACKNOWLEDGMENT

First of all we would like to express our gratitude to our parents. We would like to thank Prof. Tamal Deb for this valuable suggestion and guidance which helped us in materializing this paper. It gives us great pleasure and satisfaction that we are able to present the paper on “A Heuristic Approach to Factoid Question Generation from Sentence”. We express our gratitude towards our friends, teachers and colleagues who have directly or indirectly helped in the completion of this paper.

REFERENCES

[1] Liu, Ming, Rafael A. Calvo, and Vasile Rus. "G-Asks: An intelligent automatic question generation system for academic writing support." Dialogue & Discourse 3.2 (2012): 101-124.
[2] Chen, Wei, and Jack Mostow. "Using Automatic Question Generation to Evaluate Questions Generated by Children." The 2011 AAAI Fall Symposium on Question Generation. 2011.
[3] Radev, Dragomir, et al. "Probabilistic question answering on the web." Journal of the American Society for Information Science and Technology 56.6 (2005): 571-583.
[4] Roussinov, Dmitri, and Jose Robles. "Web question answering through automatically learned patterns." Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries. ACM, 2004.
[5] Agarwal, Manish, and Prashanth Mannem. "Automatic gap-fill question generation from text books." Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications. Association for Computational Linguistics, 2011.
[6] Skalban, Yvonne, et al. "Automatic Question Generation in Multimedia-Based Learning." COLING (Posters). 2012.
[7] Xu, Yushi, Anna Goldie, and Stephanie Seneff. "Automatic question generation and answer judging: a q&a game for language learning." SLaTE. 2009.
[8] Rus, Vasile, and C. Graesser Arthur. "The question generation shared task and evaluation challenge." The University of Memphis. National Science Foundation. 2009.
[9] Lin, Chin-Yew. "Automatic question generation from queries." Workshop on the Question Generation Shared Task. 2008.
[10] Ali, Husam, Yllias Chali, and Sadid A. Hasan. "Automation of question generation from sentences." Proceedings of QG2010: The Third Workshop on Question Generation. 2010
[11] Bird, Steven, Ewan Klein, and Edward Loper. Natural language processing with Python. " O'Reilly Media, Inc.", 2009.
[12] Kumar, Virendra, Imran Khan, and Vikas Choudhary. "A Question Generator System Using Stanford Parsing." International Journal of Engineering Research and Development 7.2 (2013): 01-05.
[13] Daelemans, W. and A. van den Bosch (2005) "Memory-Based Language Processing." Cambridge: Cambridge University Press.
[14] Question Generation as a Competitive Undergraduate Course Project Noah A. Smith, Michael Heilman, and Rebecca Hwa. In Proceedings of the NSF Workshop on the Question Generation Shared Task and Evaluation Challenge, Arlington, VA, September 2008.
[15] Jonathan Berant, Andrew Chou, Roy Frostig, Percy Liang. Semantic Parsing on Freebase from Question-Answer Pairs. Empirical Methods in Natural Language Processing (EMNLP), 2013.
[16] Bird, Steven. "NLTK: the natural language toolkit." Proceedings of the COLING/ACL on Interactive presentation sessions. Association for Computational Linguistics, 2006.