Question Generation via Overgenerating Transformations and Ranking

Michael Heilman
Language Technologies Institute
Carnegie Mellon University
mheilman@cs.cmu.edu

Noah A. Smith
Language Technologies Institute
Carnegie Mellon University
nasmith@cs.cmu.edu

Abstract

We describe an extensible approach to generating questions for the purpose of reading comprehension assessment and practice. Our framework for question generation composes general-purpose rules to transform declarative sentences into questions, is modular in that existing NLP tools can be leveraged, and includes a statistical component for scoring questions based on features of the input, output, and transformations performed. In an evaluation in which humans rated questions according to several criteria, we found that our implementation achieves 43.3% precision-at-10 and generates approximately 6.8 acceptable questions per 250 words of source text.

1 Introduction

The generation of interrogative sentences by humans has long been a major topic in linguistics, motivating various theoretical work (e.g., Ross, 1967), in particular those that view a question as a transformation of a canonical declarative sentence (Chomsky, 1973). In computational linguistics, questions have also been a major topic of study, but primarily with the goal of *answering* questions (Dang et al., 2008).

Automatic question *generation* (QG) is also a worthwhile enterprise, with applications in dialogue systems (Walker et al., 2001) and educational technologies (Graesser et al., 2005), for example. With respect to education, QG mechanisms might improve and diversify the questions posed to students by tutoring systems, leading to more natural and effective student-tutor interactions. QG might also be used with arbitrary source texts for instruction in less interactive settings.

In this paper, we focus on QG in service of practice and assessment materials for literacy instruc-

tion. Specifically, we focus on educational reading to acquire content knowledge (rather than improving reading ability). Our aim is to generate questions that assess the content knowledge that a student has acquired upon reading a text. We restrict our investigation to fact-based questions about literal information present in the text, but we believe our techniques can be extended to generate questions involving inference and deeper levels of meaning.

We follow earlier attempts to apply computational linguistics methods to QG (Mitkov et al., 2006; Kunichika et al., 2004; Gates, 2008). Our key contribution is the description and evaluation of an implemented framework that (1) is designed to be domain-general, (2) uses extensible compositions of rules to transform declarative sentences into questions, (3) explicitly handles relevant linguistic phenomena such as island constraints, (4) is modular, permitting existing NLP tools to be leveraged, and (5) includes a learned component for ranking generated questions according to various criteria.

In §2 we describe our framework for QG. §3 details our specific implementation. In §4 we describe a study in which humans rated automatically generated questions from Wikipedia and news articles. The results show that our implementation achieves 43.3% acceptability for the top 10 ranked questions (i.e., 43.3% precision-at-10). We then address related work (§5) and conclude (§7).

2 Framework

We define a framework for generating a ranked set of fact-based questions about the text of a given article. From this set, the top-ranked questions might be given to an educator for filtering and revision, or perhaps directly to a student for practice.

The generation of a single question can be usefully decomposed into the three modular stages
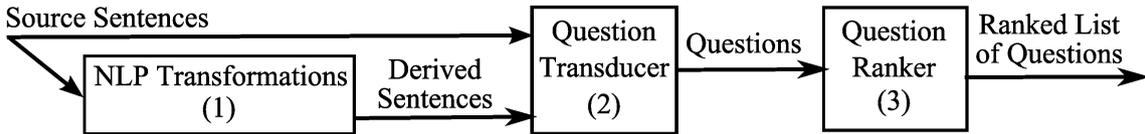
Figure 1: Three-stage framework for automatic question generation.

depicted in Figure 1.

Many useful questions can be viewed as lexical, syntactic, or semantic transformations of the declarative sentences in a text. In stage 1, a selected sentence or a set of sentences from the text is transformed into one declarative sentence by optionally altering or transforming lexical items, syntactic structure, and semantics. Many existing NLP transformations might be exploited in this stage, including extractive summarization, sentence compression, sentence splitting, sentence fusion, paraphrase, textual entailment, lexical semantics for word substitution.

In stage 2, the declarative sentence is turned into a question by executing a set of well-defined syntactic transformations (WH-movement, subject-auxiliary inversion, etc.), many of which are possible. We call this module a "question transducer."

Since different sentences from the input text, as well as different transformations of those sentences, may be more or less likely to lead to high-quality questions, in stage 3 the questions are scored and ranked according to features of the source sentences, input sentences, the question, and the transformations used in generation. Since many options are available at each stage, this is a "overgenerate-and-rank" strategy (Walker et al., 2001; Langkilde and Knight, 1998).

3 Implementation

We next present the details of our framework, implementing several operations at each of the three stages. We use techniques from summarization and sentence compression in stage 1, declarative question-generating rules motivated by research in theoretical syntax in stage 2, and a statistical reranker in stage 3.

3.1 Conventions and Definitions

The term "source sentence" refers to a sentence taken directly from the input document, to be used to generate a question (e.g., *Kenya is located in Africa.*). In contrast, the term "derived sentence" refers to a declarative sentence derived in stage 1.

"Input sentence" refers to a source or derived sentence to be given as input to the question transducer. The term "answer phrase" refers to phrases in declarative sentences which may serve as targets for WH-movement, and therefore as possible answers to generated questions (e.g., *in Africa*). The term "question phrase" refers to the phrase containing the question word replacing an answer phrase (e.g., *Where* in *Where is Kenya located?*).

We use simplified Penn Treebank-style phrase structure trees, including POS and category labels, as produced by the Stanford Parser (Klein and Manning, 2003) to represent the syntactic structure of sentences. Noun phrase heads are selected using Collins' rules (Collins, 1999).

3.2 Searching and Manipulating Trees

In order to implement the rules for transforming source sentences into questions, we use `Tregex`, a tree query language, and `Tsurgeon`, a tree manipulation language built on top of `Tregex` (Levy and Andrew, 2006). These languages were also employed for QG by Gates (2008), though with a different approach (cf. §3.4).

The `Tregex` language includes various relational operators based on the primitive relations of immediate dominance (denoted "<") and immediate precedence (denoted "."). In addition to supporting queries involving standard tree relations, `Tregex` also includes some linguistically motivated constraints such as headship, constrained dominance, and precedence (e.g., dominance through a chain of verb phrases). `Tsurgeon` adds the ability to *modify* trees by relabeling, deleting, moving, and inserting nodes.

Queries in `Tregex` find nodes that match patterns; these nodes may be named and manipulated using `Tsurgeon` operations. For example, the `Tregex` expression "SBAR < /^WH.*P$/ << NP|ADJP|VP|ADVP|PP=unmv" would find a node labeled as one of a set of phrase types ("NP|ADJP|VP|ADVP|PP") which is dominated by a clause ("SBAR") that also *directly* dominates a question phrase ("/^WH.*P\$/"). This node can be retrieved through its user-assigned

name, "unmv" in this case. This rule is used to identify the constraint on movement in clauses such as *whether John likes Mary*.

The rules in our system are mostly implemented using Tregex and Tsurgeon. In some cases, however, we resort to transformations outside the expressive power of Tsurgeon, which cannot perform operations one at a time (all matching nodes must be transformed simultaneously), back-reference relabeled nodes, or include reserved words (e.g., "insert") in new node labels.

3.3 Transforming Source Sentences (Stage 1)

We explored several transformations for stage 1, which takes as input the original text and produces declarative derived sentences to be further modified by the question transducer (§3.4).

An analysis of the question transducer's output when applied directly to sentences from source texts revealed that complex sentences often lead to unnatural or even senseless questions. Therefore, various simplifying transformations are performed in stage 1 to remove phrase types such as leading conjunctions, sentence-level modifying phrases, and appositives. Such transformations have been utilized in previous work on headline generation (Dorr and Zajic, 2003) and summarization (Toutanova et al., 2007). Table 1 describes these transformations.

Further, the question transducer will not generate questions about certain kinds of syntactically embedded content (cf. Table 3). Questions can be produced about much of this embedded content if we extract declarative sentences from finite clauses, relative clauses, appositives, participial phrases. The sentence transformation module implements these options. For example, it transforms the sentence *Selling snowballed because of waves of automatic stop-loss orders, which are triggered by computer when prices fall to certain levels* into *Automatic stop-loss orders are triggered by computer when prices fall to certain levels*, from which stage 2 produces *What are triggered by computer when prices fall to certain levels?*. Table 2 describes the expressions and rules for extraction.

We allowed for some of the operations in this first stage to be fairly complex because they are less central to the whole system (for many sentences, these stage 1 rules do not fire at all). In contrast, the rules in stage 2, most of which are utilized for every generated question, maintain a higher degree of simplicity and elegance.

3.4 Question Transducer (Stage 2)

In stage 2 of our implementation, the question transducer takes as input a declarative sentence and produces as output a set of possible questions.[1] It identifies the answer phrases which may be targets for WH-movement and converts them into question phrases. In the current system, answer phrases can be noun phrases or prepositional phrases, which enables *who*, *what*, *where*, *when*, and *how much* questions.

While many of the answer phrases turn out to be valid answers to the generated questions, some exhibit referential ambiguity (e.g., *the planet* in a text on the solar system). We leave the generation of correct answers and distractors to future work.

The system could be extended to detect and transform other types of phrases to produce other types of questions (e.g., *how*, *why*, and *what kind of*). It should be noted that the transformation from answer to question is achieved through a composition of general-purpose rules. This would allow, for example, the addition of a relatively simple rule to generate *why* questions by building off of the existing rules for subject-auxiliary inversion, verb decomposition, etc. In contrast, previous QG approaches have employed separate rules for specific sentence types (e.g., Mitkov and Ha, 2003; Gates, 2008).

For each sentence, many questions may be produced: there are often multiple possible answer phrases in a particular sentence, and multiple question phrases for each answer phrase. For example, from the sentence *Francium was discovered by Marguerite Perey in France in 1939*,[2] the transducer produces the following questions:

- *Where was francium discovered by Marguerite Perey in 1939?*

- *When was francium discovered by Marguerite Perey in France?*

- *Was francium discovered by Marguerite Perey in France in 1939?*

- *By what was francium discovered in France in 1939?*

---

[1]The main clauses of declarative sentences are marked as "S" in the Penn Treebank. We leave inverted clauses ("SINV") for future work.

[2]From "Francium." *Wikipedia: The Free Encyclopedia.* Retrieved Dec. 16, 2008.

Description of Transformation	Expression
Sentence-initial conjuctions are removed by deleting `conj`.	`ROOT < (S < CC=conj)`
Sentence-initial adjunct phrases are removed by deleting `adjunct`. (A nearly identical rule deletes commas following these adjuncts.)	`ROOT < (S < (/[^,]/=adjunct $.. (/,/ $.. VP)))`
Appositives are removed by deleting `app`, `lead`, and `trail`. (A nearly identical rule deletes parenthetical phrases.)	`SBAR|VP|NP=app $, /,/=lead $. /,/=trail !$ CC !$ CONJP`

Table 1: Rules used in stage 1 to simplify and compress sentences.

Description of Transformation	Expression
A new tree is constructed from an appositive phrase `app` modifying a noun phrase `noun` by creating a sentence of the form `noun copula app`, where `copula` is in the past tense and agrees with the head of `noun`.	`NP !< CC !< CONJP < (NP=noun $.. (/,/ $.. (NP=app $.. /,/)))`
A new tree is constructed from a finite clause by placing `finite` under a new root node, with the appropriate punctuation `punct`. There is a check to avoid extracting clauses dominated by noun phrases or prepositional phrases. (Extracting phrases in such cases too often led to vague or ungrammatical questions during development.)	`S=finite !>> NP|PP < NP < (VP < VBP|VB|VBZ|VBD|MD) ?< /\\./=punct`
A new tree is constructed from a verbal modifier (e.g., *bought by John* in *This is the car bought by John.*) by creating a sentence of the form `noun copula modifier`, where `copula` is in the past tense and agrees with the head of `noun`.	`NP=noun > NP $.. VP=modifier`
A new tree is constructed from a relative clause by creating a sentence in which `noun` is the subject or object of the clause in `rel` that is missing a subject or object. `relclause` is recursively searched to find a clause missing a noun phrase or a verb phrase missing a noun phrase. The noun phrase modified by the relative clause is assumed to be the subject or object of that clause even though noun phrases are not marked in the parse trees. The main clause of the newly extracted sentence is often `relclause` itself, but not always (e.g., *Mary met the man* would be extracted from *The man John said Mary met.*)	`NP=noun > NP $.. (SBAR < S=rel !< WHADVP !< WHADJP)`

Table 2: Rules used in stage 1 to extract simple sentences from more complex ones.

Of course, the last question ought to be *By whom was....* This error is due to the failure of the entity recognition component, discussed in §3.4.2, to correctly identify *Marguerite Perey* as a person. Thus, this example also illustrates the importance of ranking to avoid questions whose features are strongly associated with errors (stage 3, §3.5).

The question transducer aims to overgenerate grammatical, though perhaps irrelevant or unimportant, questions. These rules encode a substantial amount of linguistic knowledge. They (1) mark phrases that cannot be answer phrases, due, for example, to island constraints; (2) remove each answer phrase and generate possible question phrases for it; (3) decompose the main verb; (4) invert the subject and auxiliary verb; and (5) insert one of the question phrases. (Note that some of these steps do not apply in some cases.) We discuss each of these in turn.

Next we described the sequence of operations which the system performs to convert declarative sentences into questions. This process is illustrated in Figure 2.

3.4.1 Marking Unmovable Phrases

A set of `Tregex` expressions marks the phrases in an input tree which cannot be answer phrases due to constraints on WH-movement. Each of those tree nodes is renamed by extracting the node's current label, prepending a special marker on the label ("UNMV-"), and relabeling the node with a `Tsurgeon` operation. The expressions are listed and described in Table 3.

The operations at this step occur in parallel (i.e., their ordering does not matter) and can therefore easily be extended or improved upon. There are two exceptions to the parallel operation of rules which serve to propagate constraints down the trees. These two rules are applied after all others. The first marks as unmovable all nodes under an unmovable node. The second marks as unmovable all nodes under an otherwise *movable* node, which encodes the fact that noun phrases are islands to movement.

In particular, the constraint which encodes that noun phrases are islands to movement ensures, for example, that phrases in relative clauses cannot undergo movement. For example, the spurious

Constituents to mark as unmovable	Expression
Adjunct clauses under verb phrases. Such phrases typically follow commas.	`VP < (S=unmv $,, /,/)`
Clause-level modifiers, which are any nodes directly under a clausal, or "S", node except nouns and verbs.	`S < PP|ADJP|ADVP|S|SBAR=unmv`
Phrases under conjunctions. A single conjoined node cannot undergo WH-movement while its siblings do not (e.g., *Who did John meet and Mary?* from *John met Bob and Mary.*).	`/\\.*/ < CC <<` `  NP|ADJP|VP|ADVP|PP=unmv`
Noun phrases in adjunct clauses, assuming subordinate clauses with an explicit complementizer other than *that* are adjuncts (e.g., *whether* a recession *is on the horizon*).	`SBAR < (IN|DT < /[^that]/)` `  << NP|PP=unmv`
Phrases under a question phrase. This constraint is related to the idea of subjacency (Chomsky, 1973), as is the next.	`SBAR < /^WH.*P$/` `  << NP|ADJP|VP|ADVP|PP=unmv`
The subject of a complement phrase when an explicit complementizer is present (e.g., *Who did John say that is old?* from *John said that Bob is old.*).	`SBAR <, IN|DT` `  < (S < (NP=unmv !$,, VP))`
Phrases under unmovable nodes. This propagates the constraints down the tree.	`@UNMV << NP|ADJP|VP|ADVP|PP=unmv`
Any nodes under an otherwise movable phrase. This encodes the constraint that noun phrases are islands to movement.	`NP|PP|ADJP|ADVP` `  << NP|ADJP|VP|ADVP|PP=unmv`
Optional expressions for conservative restrictions on the set of possible answer phrases:	
Prepositional phrases that do not have a noun phrase object (e.g., *on seeing an old friend*).	`PP=unmv !< NP`
Both of a pair of noun phrases that are siblings in the tree (e.g., to avoid *Who did John give a book?* from *John gave* Mary a book.*).	`S < (NP=unmv $ NP|UNMV-NP)`
Existential there noun phrases. This ensures against spurious questions such as "What was a dog in the park?" being generated from *There was a dog in the park.*	`NP=unmv < EX`

Table 3: Tree searching expressions for identifying phrases which may not undergo WH-movement. The prefix `UNMV-` identifies nodes already marked as unmovable.

Question Phrase	Conditions
Who	a noun phrase whose head is labeled PERSON, PER_DESC, or ORGANIZATION
Where	a noun phrase whose head is labeled LOCATION, or is a prepositional phrase with certain prepositions (*in*, *at*, *on*, *over*) whose head is labeled LOCATION
When	a noun phrase whose head is labeled DATE or TIME, or is a prepositional phrase with certain prepositions (*in*, *at*, *on*, *over*) whose head is labeled DATE or TIME
How much	a noun phrase with a quantifier phrase or word ("QP" or "CD") and whose head is labeled MONEY
A preposition followed by any of the above	a prepositional phrase whose object is a noun phrase that satisfies one of the above conditions

Table 4: Conditions for generating questions with certain question phrases.

question *Who did I buy the book that inspired?* will not result from *I bought the book that inspired Bob.* It also disallows movement of prepositional phrase objects, in order to avoid two very similar questions and to simplify the rules (e.g., *To whom did I give the book.* and *Whom did I give the book to.*).

3.4.2 Generating Possible Question Phrases

After marking unmovable phrases, the transducer iterates over the possible answer phrases. For each one, it copies the input tree, then removes the answer phrase and generates possible question phrases from it. (This step is skipped for yes-no questions.) [3]

The question phrases for a given answer phrase consist of a question word (e.g., *who*, *what*, *where*, *when*), possibly preceded by a preposition and, in the case of question phrase like *whose car*, followed by the head of the answer phrase.

To generate the question phrases, the system annotates the source sentence with a set of entity types taken from the BBN Identifinder Text Suite (Bikel et al., 1999). The set of labels from BBN includes those used in standard named entity recognition tasks (e.g., "PERSON," "ORGANIZATION" as in the MUC-6 Named Entity Task) and their corresponding types for common nouns (e.g., "PER_DESC," "ORG_DESC"). It also includes dates, times, monetary units, and others.[4]

---

[3] This process allows us to extract a potential answer along with each question. However, we leave the exploration and

evaluation of extracting answers for future work.

[4] We use the open-source Stanford NER tool (Finkel et al.,

Purpose	Expression
To identify the main verb for decomposition into a form of "do" and the base form.	`ROOT < (S=clause < (VP=mainvp < /VB.?/=tensed !< (VP < /VB.?/))).`
To identify the main clause for subject-auxiliary inversion.	`ROOT=root < (S=clause <+(/VP.*/) (VP < /(MD|VB.?)/=aux < (VP < /VB.?/=verb)))`
To identify the main clause for subject auxiliary inversion in sentences with a copula and no auxiliary (e.g., *The currency's value is falling*).	`ROOT=root < (S=clause <+(/VP.*/) (VP < (/VB.?/=copula < is|are|was|were|am) !< VP))`

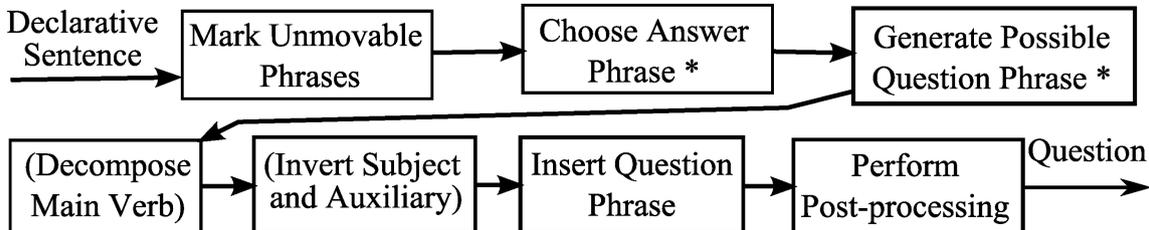Table 5: Tree searching expressions in the question transducer.

Figure 2: Process of transforming declarative sentences into questions in the Question Transducer (stage 2). Parentheses mark steps which may not be necessary for certain questions. ∗ mark steps which may produce zero or multiple outputs.

For a given answer phrase, the system uses the phrase's entity labels and syntactic structure to generate a set of zero or more possible question phrases, each of which is used to generate a final question sentence. Table 4 describes the conditions under which each type of question phrase is produced.

3.4.3 Decomposition of the Main Verb

In order to perform subject-auxiliary inversion (§3.4.4), if an auxiliary verb or modal is not present, the question transducer decomposes the main verb into the appropriate form of *do* and the base form of the main verb. It then modifies the tree structure of the verb phrase accordingly. For example, *John saw Mary* becomes *John did see Mary* before transformation into *Who did John see?* rather than **Saw John Mary?*.

If an auxiliary verb is already present, however, this decomposition is not necessary (e.g., *John has seen Mary.* could lead to *Who has John seen?*). In such cases, the main verb phrase includes the auxiliary verb and a nested verb phrase containing the base form of the main verb.

The system identifies main verbs that need to be decomposed with the `Tregex` expression shown at the top of Table 5.

2005) to perform the annotation. We retrained the Stanford NER model on 102 texts that were labeled by the proprietary BBN Identifinder tool, essentially building a CRF model that predicts the tags, albeit with noisy data. The training texts were randomly sampled from the set of featured articles on Wikipedia as of Dec. 16, 2008. Performance was not evaluated rigorously but judged adequate on manual inspection.

In order to convert between lemmas of verbs and the different surface forms that correspond to different parts of speech, we created a map from pairs of verb lemma and part of speech to verb surface forms. We extracted all verbs and their parts of speech from the Penn Treebank (Marcus et al., 1993). We lemmatized each verb first by checking morphological variants in WordNet (Miller et al., 1990), and if a lemma was not found, then trimming the rightmost characters from the verb one at a time until a matching entry in WordNet was found. This simple approach works in practice for English because most verb forms either are derived from the lemma by adding a few letters (e.g., "ed") or are mapped to lemmas in WordNet's database.

3.4.4 Subject-Auxiliary Inversion

The transducer performs subject-auxiliary inversion either when the question to be generated is a yes-no question or when the answer phrase is a non-subject noun phrase. The bottom two `Tregex` expressions in Table 5 identify the main clause and the relevant nodes in the verb phrase.

The main clause node, initially "S", is first relabeled as "SQ", indicating that it is part of a question. Then the auxiliary or copula is moved so that it becomes the first child. The "SQ" node is then used to form a new tree for the sentence. In the case of yes-no questions, the root node of the question tree will have the "SQ" node as its only child. In the case of WH-questions, the root node has an "SBARQ" node as its child. This "SBARQ"

node then has the "SQ" node as a child, which will be preceded by a question phrase node (e.g., "WHNP").

After transforming the main clause and relabeling it "SQ," any leading adjunct phrases under this node and preceding a comma are moved to the front of the final sentence (e.g., to produce more natural questions like *Following Thomas Jefferson, who was elected the 4th president?* rather than the more awkward *Who, following Thomas Jefferson, was elected the 4th president?*).

3.4.5 Inserting Question Phrases

Each possible question phrase is inserted into a copy of the tree to produce a question. The question phrase is inserted as a child of the "SBARQ" node under the root node, following any leading sentence-level adjunct phrases. If the question is a yes-no question, then this step is not necessary.

3.4.6 Post-processing

Some additional post-processing mechanisms are necessary to ensure proper formatting and punctuation. Sentence-final periods are changed to question marks. Additionally, the output is detokenized to remove extra whitespace (e.g., preceding punctuation symbols).

We observed in preliminary output of our system that nearly all of the questions including pronouns were too vague (e.g., *What does it have as a head of state?* from a Simple English Wikipedia article on Thailand.). We therefore filtered all questions with personal pronouns, possessive pronouns, and noun phrases consisting solely of determiners (e.g., *those*), which eliminated a substantial portion of the possible questions our system might output. This suggests a possible future extension to leverage coreference resolution and referring expression generation to replace pronouns with referring expressions.

3.5 Question Ranker (Stage 3)

The question transducer and the sentence transformation stages overgenerate, producing erroneous questions for various reasons. We already noted an example related to entity recognition in §3.4.

Another example is that errors during automatic parsing may propagate. If the sentence *Bob donated the book in his backpack to the library* were parsed such that the phrase *in his backpack* attached to the verb *donated* rather than *book*, the

erroneous question *What did Bob donate in his backpack to the library?* would result.

The sentence transformation module in stage 1 is particularly prone to errors. For example, it does not consider negation or semantics when extracting finite clauses, and thus for the sentence *John never believed that Hamilton shot Aaron Burr*, it suggests the misleading question *Who shot Aaron Burr?*[5]

That certain features of the input and certain operations are more strongly associated with errors suggests that scoring questions by some acceptability measure may allow us to rank them effectively. In this section, we describe our implementation of a module to score and rank questions.

3.5.1 Model

We use a discriminative reranker (Collins, 2000), specifically based on a logistic regression model that defines a probability of acceptability, given the question q and source text t: $p(\cdot \mid q, t)$. We use a to denote "acceptable" and u to denote "unacceptable"; $p(\mathsf{u} \mid q, t) = 1 - p(\mathsf{a} \mid q, t)$.

Questions may be deficient, or unacceptable, in various ways. In Table 6, we enumerate a set of possible deficiencies that humans may be able to identify. In our first ranking approach ("Boolean"), we collapse the possible deficiencies so that questions with any of the possible deficiencies are treated as unacceptable, and learn a logistic regression model over $\{\mathsf{a}, \mathsf{u}\}$.

In our second approach ("Aggregate"), we learn separate conditional models of the probability of a given question being acceptable according to each of the deficiencies in Table 6, then combine them:

$$p(\mathsf{a} \mid q, t) = \prod_{i=1}^{K} p_i(\mathsf{a}_i \mid q, t) \qquad (1)$$

where i indexes different types of acceptability (with respect to grammaticality, making sense, vagueness, etc.), and K is the number of types, 8 in our case.

3.5.2 Features and Parameter Estimation

Both ranking models use the same features, which are listed in Table 7.

We do not claim that this is the optimal set of features. That is, the ranking model could certainly be improved by refining the feature set. In particular, features based on n-gram language models of questions might provide useful information if an appropriate corpus of questions could be

---

[5]In fact, Aaron Burr shot Alexander Hamilton.

Question Deficiency	Description
Ungrammatical	The question does not appear to be a valid English sentence.
Does not make sense	The question is grammatical but indecipherable. (e.g., *Who was the investment?*)
Vague	The question is too vague to know exactly what it is asking about, even after reading the article (e.g., *What did Lincoln do?*).
Obvious answer	The correct answer would be obvious even to someone who has not read the article (e.g., the answer is obviously the subject of the article, or the answer is clearly *yes*).
Missing answer	The answer to the question is not in the article.
Wrong WH word	The question would be acceptable if the WH phrase were different (e.g., *in what* versus *where*). WH phrases include *who, what, where, when, how, why, how much, what kind of*, etc.
Formatting	There are minor formatting errors (e.g., with respect to capitalization, punctuation)

Table 6: Possible deficiencies a generated question may exhibit.

Feature
The numbers of tokens in the source sentence, question sentence, and answer phrase.
The mean unigram language model probabilities of the source sentence and answer phrase.
Binary variables indicating whether the input sentence was derived from a relative clause, a participial phrase, an appositive phrase, or a simplified version of the source sentence.
A binary variable indicating the use of each possible WH word, or whether the question was a yes-no question.
The parse log likelihood of the source sentence, as well as the log likelihood normalized by length.
Binary variables indicating the presence of different parts of speech in the answer phrase.
A binary variable indicating whether the answer phrase is a prepositional phrase or a noun phrase.
A binary variable indicating whether the answer phrase is the subject in the input sentence (e.g., requires do-inversion).
Binary variables indicating the main predicate's tense (past, present, future).
The number of noun phrases in the question.
The number of prepositional phrases in the question.
A binary variable indicating the presence of negation words in the source sentence (*not, no*, and *never*).

Table 7: Features for ranking questions.

found. We did not explore this particular type of feature for our experiments because from observations of preliminary output, it appeared that most syntactic errors were related to longer distance dependencies than what would be modeled by an n-gram model. The ranker could also include features for specific constructions that might appear in input sentences, or even features tailored to a particular QG application.

We estimate the parameters by optimizing the regularized log-likelihood of the training data (cf. §4.1), with the regularization constant selected through cross-validation.

4 Evaluation

No "standard" evaluation task yet exists for QG. To evaluate our implemented QG framework, we conducted an experiment in which 15 native English-speaking university students rated the system's output, indicating whether each question exhibited any of the deficiencies listed in Table 6.[6]

Annotators were asked to read the text of an article and then rate approximately 100 questions generated from the text. They worked in a web-based interface and could re-read the article and alter any of their previous ratings as they saw fit. They were asked to consider each question independently, such that similar questions about the same information would receive similar ratings.

Three people rated each of the questions in the test set. Since the problem deficiencies are not mutually exclusive, to estimate inter-rater agreement we computed separate Fleiss's κ values for each deficiency. The values are given in Table 8. The κ value of 0.42 for the "Acceptable" category corresponds to "moderate agreement"(Landis and Koch, 1977). The agreement is lower than other rating schemes,[7] due in part to the rating scheme but also to the fact that the raters were novices (note, for instance, that κ is only 0.495 for "formatting"). However, the fact that 61.5% of the test-set question ratings for the "acceptable" category were unanimous gives some confidence.

Primarily, we report results based on majority ratings (i.e., the rating assigned by 2 of the 3 raters). However, since agreement was moderate, and since, in spot-checks, we observed that raters appeared more likely to liberally accept bad questions than to reject good ones, we also report results in which a question is deemed to have a particular deficieny if *any* of its three raters labeled it as having that deficiency. This second set of

---

[6]The ratings from one person were excluded due to an extremely high rate of accepting questions as error-free and other irregularities.

[7]E.g., Dolan and Brockett (2005) and Glickman et al. (2005) report κ values around .6 for paraphrase identification and textual entailment, respectively.

measurements provides us with an estimate of the lower bound on the quality of generated questions.

In addition to the test set, we created a training data set for learning to rank questions. In the training set, each article's questions were rated by only one person.

4.1 Corpora

The training and test data sets consisted of questions about articles from 4 corpora.

One corpus, (WIKI-ENG) was a random sample from the featured articles in the English Wikipedia[8] that had between 250 and 2,000 word tokens. This English Wikipedia corpus provides expository texts written at an adult reading level from a variety of domains, which roughly approximates the prose that a secondary or post-secondary student would encounter. By choosing from the featured articles, we intended to select well-edited articles about topics of general interest. The test set included 137 questions about 2 articles from WIKI-ENG. The training set included 1,352 questions about 12 articles.

A second corpus (WIKI-SIMP) was a random sample from the articles in the Simple English Wikipedia of similar length. This corpus provides similar text but at a reading level corresponding to elementary education or intermediate second language learning.[9] While the WIKI-SIMP corpus contains articles with shorter sentences, which we would expect to make processing easier, it also contains more errors than the relatively well-edited WIKI-ENG articles. The test set included 125 questions about 2 articles from WIKI-SIMP. The training set included 1,241 questions about 16 articles.

The third and fourth corpora were from Section 23 of the *Wall Street Journal* data in the Penn Treebank (Marcus et al., 1993). While these articles are somewhat different in genre (news) and domain (mostly business and politics) from our focus, they allow for an experimental study of the effects of the accuracy of syntactic parsing on the quality of automatically generated questions. We used the same articles for both of these corpora to control for subject matter. For one corpus (WSJ), we used the Stanford Parser to derive parse trees, while for the other (WSJ*), we used the human-annotated gold-standard parse trees. The test set included 190 questions about 2 articles from WSJ, and 192 questions about 2 articles from WSJ*. The training set included 284 questions about 8 articles from WSJ. No rater saw questions about the same article from both WSJ and WSJ*.

4.2 Results for Unranked Questions

First, we present results for the unranked questions produced by Stages 1 and 2. As shown in Table 8, 31.5% of questions were labeled by the majority of raters as having no deficiencies. 12.8% of the questions from all 4 corpora, including WSJ*, were labeled by all 3 raters as having no deficiencies.

The most frequent deficiency, exhibited by 18.7% of questions, is vagueness (e.g., *Who was Gerald Ford?* from *Nixon's second vice president was Gerald Ford.*). Ungrammaticality (15.4%) and semantic errors ("No sense" at 18.4%) are also quite frequent. The substantial percentage of formatting errors (8.8%) is due to both straightforward issues with pre-processing the articles and more challenging issues such as failing to identifying named entities (e.g., *Who was nixon's second vice president?*).

While Table 8 provides some measure of precision, recall would require knowing the number of possible valid questions. Instead, we provide a measure of *productivity*: according to majority ratings, stages 1 and 2 produced an average of 6.8 acceptable questions per 250 words (i.e., approximately one page of text in a printed book). According to the standard of unanimous acceptance of questions, stages 1 and 2 produced 3.1 acceptable question per 250 words.

Table 9 provides some examples of questions produced by stages 1 and 2, in order to illustrate some of the strengths and weaknesses of our implementation. We observed that in many cases these errors are the result of incorrect automatic parsing and entity labeling.

4.3 Results for Ranking

To evaluate ranking methods, we calculate the percentage of acceptable questions in the top N questions, or precision-at-N. We employ this metric because a typical user would likely consider only a limited number of questions.

Deficiency:	Ungram.	No sense	Vague	Obvious	Missing	Wrong WH	Format	Other	Acceptable
Majority (%):	15.4	18.4	18.7	1.7	0.9	5.1	8.8	0.9	31.5
Any Rater (%):	36.7	39.5	40.2	14.7	5.1	12.2	18.3	10.8	12.8
Fleiss's κ:	0.29	0.29	0.40	0.13	0.14	0.40	0.50	0.03	0.42

Table 8: Percentages of questions generated from the test set for which raters indicated a deficiency, averaged across articles and corpora, and inter-rater agreement. The "Majority" row includes the percentages of questions for which a majority (i.e., 2) of the raters indicated a deficiency. The "Any Rater" row includes the percentages of questions for which any one of the raters indicated a deficiency. The rightmost column is for the distinction between exhibiting none or any of the deficiencies.

Table 11 shows results for the binary and aggregate ranking methods, along with percentages without ranking for comparison. For the automatically parsed corpora, 26.6% of *all* questions were rated as acceptable by a majority of raters (Note that the overall percentage in §4.2 is different because it includes questions from WSJ*). While boolean ranking did not appreciably improve percentage of acceptable top-ranked questions, aggregate ranking led to 43.3% precision-at-10 and 40.0% precision-at-25.

Looking at per-corpus results, we observe that in some cases such as WIKI-SIMP, the precision-at-N values counterintuitively *increase* as N increases. This can be attributed to the fact that these values are based on only two articles, and the precision-at-N metric is unstable at low N when there are a very small number of articles. For example, the precision-at-1 would be expected to vary widely since it depends only on the two most highly ranked questions. In order to improve the stability of this metric, one can either use higher values of N or compute the metric across more articles.

To test whether the mean, computed across the 6 automatically parsed articles, of the observed precision-at-25 value from the Aggregate ranking approach was statistically significantly better than chance, we sampled from the distribution of precision-at-25 values corresponding to the null hypothesis that the rankings are just random orderings of the questions. For 100,000 iterations, we randomly ranked the questions for each article and computed the mean precision-at-25 across articles. Since the proportion of samples exhibiting a value more extreme than our observed mean precision-at-25 of 40.0% was 0.00001 (i.e., $p < 0.05$), we have sufficient evidence to reject the null hypothesis. We repeated this test for precision-at-10, and the proportion of random orderings with higher precision-at-10 values than our observed value of 43.4% was 0.00184 (i.e., $p < 0.05$).

We also estimated an upper bound for ranking performance based on the chance of one rater's accepting a question agreeing with other raters' judgments. We define our estimate as the conditional probability of all human raters annotating a question as acceptable given that *one* of the raters annotated that question as acceptable. If r_{ij} is a human rating from rater i for question j, a indicates a rating of "acceptable", r_{1j} through r_{Nj} are the full set of N ratings for question j, then the upper bound is $P(r_{1j} = ... = r_{Nj} = p | r_{ij} = p)$. The observed data yield an estimate of 59.5%. Note that this upper bound is based on binary judgments rather than scores assigned by humans. Therefore, it is the same across different numbers of questions considered (e.g., the top-10 or top-25). Also, to provide estimates at rank N, it relies on the assumption that at least N good questions exist for each article, which may not be valid as N grows large.

It is notable that 41.5% of the questions about the WSJ* articles were rated as acceptable, with 52.0% precision-at-25 after aggregate ranking, compared to 24.8% overall and 36.0% precision-at-25 for WSJ. These results suggest that the many of the unacceptable questions are generated due to errors in automatic parsing.

Surprisingly, fewer of the questions from WIKI-SIMPwere judged to be acceptable. This is likely in part due to the less well-edited nature of the Simplified English Wikipedia (e.g., the following sentence from an article about the Berlin Blockade after World War II: *Too keep everything safe , air traffic control located at Tempelhof [sic].*), as well as the awkward grammatical constructions which are prevalent in the corpus due to efforts to limit the length of sentences. However, we hesitate to make strong conclusions about differences between corpora based on two pairs of articles on different topics.

5 Related Work

Several taxonomies of questions have been developed which may help to guide the study of

Annotation	Question	Source Sentence	Comments
Acceptable	*What is the traditional religion of Japan?*	*Shinto is the traditional religion of Japan and some consider many of Hayao Miyazaki's movies–including Totoro–to have Shintoist themes.*	Many questions are successfully extracted from the main clauses of sentences.
Acceptable	*What were badly damaged from the Great Hanshin earthquake?*	*The race, originally scheduled to be held as the third round of the season on April 16, 1995, was moved to October as the local infrastructure and communications were badly damaged from the Great Hanshin earthquake.*	As in this example about the Pacific Grand Prix automotive race, the system successfully produces questions from subordinate clauses by applying the transformations in Stage 1.
Acceptable	*Who was deprived of both the knighthood and the earldom after taking part in the Jacobite rising of 1715?*	*But this has happened only once, to John Erskine, 6th Earl of Mar who was deprived of both the knighthood and the earldom after taking part in the Jacobite rising of 1715.*	Questions are extracted from relative clauses by applying transformations in Stage 1.
Ungrammatical	*In what were nests excavated exposed to the sun?*	*A clutch of anywhere from 1 to 21 eggs are usually laid in June or July depending on the size and age of the female, in nests excavated in pockets of earth exposed to the sun.*	The parser incorrectly attaches *exposed to the sun* to the verb phrase headed by *excavated* rather than the noun phrase *pockets of earth*. Correcting this attachment would lead to the following more sensible, if not perfect, question: *In what were nests excavated?*.
Does not make sense	*En what was Mexican General Mart n Perfecto de Cos?*	*In September, Texians began plotting to kidnap Mexican General Martín Perfecto de Cos, who was en route to Goliad to attempt to quell the unrest in Texas.*	The system does not recognize idiomatic expressions such as *en route*, which is identified as a prepositional phrase here.
Vague	*What do modern cities also have?*	*These giant cities can be exciting places to live, and many people can find good jobs there, but modern cities also have many problems.*	Questions are frequently generated about phrases with low information content such as *many problems*.
Too Easy	*Did the company say the improvement is related to additional cogeneration facilities that have been put into operation?*	*The company said the improvement is related to additional cogeneration facilities that have been put into operation.*	Overly detailed yes-no questions are often generated directly from the entire source sentence.
Missing answer	*Who were citizens of that city?*	*Some city-states were monarchies, others elected (part of) the people who governed by (part of) the people who were citizens of that city, and who lived there.*	Occasionally, it is not clear to what questions refer, and this may lead to questions being labeled as not having an answer in the text.
Wrong WH word	*In what did lockheed Martin (once Martin Marietta) open a manufacturing site in 1956?*	*Lockheed Martin (once Martin Marietta) opened a manufacturing site in Orlando in 1956.*	The entity labeling component other incorrectly identifies or fails to identify locations (e.g., *Orlando*), persons, and other entities, resulting in incorrect WH words.
Formatting	*What is carcassonne?*	*Carcassonne is an ancient city in France.*	The entity labeling component often fails to identify sentence-initial named entities for which capitalization should be preserved.

Table 9: Example questions produced by stages 1 and 2, with annotations they received during the evaluation of the system.

Corpus	Ranking	N=10	N=25	All
WIKI-ENG	Boolean	30.0	24.0	31.9
	Aggregate	50.0	52.0	31.9
WIKI-SIMP	Boolean	20.0	26.0	23.2
	Aggregate	35.0	32.0	23.2
WSJ cline2-5	Boolean	25.0	20.0	24.8
	Aggregate	45.0	36.0	24.8
WSJ*	Boolean	45.0	38.0	41.5
	Aggregate	55.0	52.0	41.5
Average (not incl. WSJ*)	Boolean	25.0	23.3	26.6
	Aggregate	43.3	40.0	26.6

Table 10: Percentages of questions rated as acceptable for each data set in the top N questions for the two ranking methods, using *majority* ratings (i.e., at most 1 of the raters indicated any deficiencies). The percentages of all unranked questions are shown in the rightmost column. Averages across the three automatically parsed corpora are shown in the bottommost row.

Corpus	Ranking	N=10	N=25	All
WIKI-ENG	Boolean	15.0	10.0	12.4
	Aggregate	20.0	20.0	12.4
WIKI-SIMP	Boolean	10.0	14.0	9.6
	Aggregate	5.0	6.0	9.6
WSJ	Boolean	10.0	8.0	11.1
	Aggregate	30.0	24.0	11.1
WSJ*	Boolean	15.0	16.0	18.3
	Aggregate	20.0	24.0	18.3
Average (not incl. WSJ*)	Boolean	11.7	10.7	11.0
	Aggregate	18.3	16.7	11.0

Table 11: Percentages of questions rated as acceptable for each data set in the top N questions for the two ranking methods, requiring *unanimous* acceptance of questions (i.e., none of the raters indicating any deficiencies). The percentages of all unranked questions are shown in the rightmost column. Averages across the three automatically parsed corpora are shown in the bottommost row.

QG (Lehnert, 1978; Schank, 1986; Harabagiu et al., 2002; Beck et al., 1997), though these focus on logical (e.g., whether inference is necessary) or psychological characteristics (e.g., is world knowledge activated) rather than linguistic ones (e.g., lexical overlap, similar constructions or transformations).

In computational linguistics, question answering has been the driving application for research on questions (Dang et al., 2008). Models of the transformation from answers to questions have been developed (Echihabi and Marcu, 2003; Wang et al., 2007; har, 2005), with the goal of finding correct answers *given* a question (i.e., in a source-channel framework). Other research has focused on retrieval or extraction (e.g., Ravichandran and Hovy, 2001; Hovy et al., 2001).

QG is a kind of natural language generation, which is often divided into content determination, discourse planning, sentence aggregation, lexicalization, referring expression generation, and linguistic realization (Reiter and Dale, 1997). We focus mainly on content determination and realization.

Much of the natural language generation research pertaining to QG has focused on gathering information from the users of dialog systems for trip planning and similar tasks (e.g., Walker et al., 2001). However, the overgenerate-and-rank approach we employ has been applied previously for generation (Walker et al., 2001; Langkilde and Knight, 1998) and parsing (Collins, 2000). A recent NSF workshop[10] had as its aim the formulation of a shared task on QG.

Many researchers have investigated the use of NLP techniques for other types of assessment and practice in the area of literacy education. For example, Brown et al. (2005) discuss various approaches to QG for vocabulary assessment and practice. Also, Mostow et al. (2004) explore the use of cloze, or fill-in-the-blank, questions for assessing reading comprehension.

Our wide-coverage approach relies on operationalizing linguistic constraints related to WH-movement widely noted in the literature. These question-related phenomena have been studied extensively. In a seminal dissertation, Ross (1967) described many of these phenomena, and in doing so provided motivation for a variety of subsequent theoretical explanations. Goldberg (2006)

---

[10]http://www.questiongeneration.org

provides a concise summary of these constraints.

Our wide-coverage approach relies on operationalizing linguistic constraints related to WH-movement widely noted in the literature. In a seminal dissertation, Ross (1967) described many of these phenomena. Goldberg (2006) provides a concise summary of them.

Previous research has also approached the topic of automatic QG directly (Mitkov and Ha, 2003; Kunichika et al., 2004; Gates, 2008). Mitkov and Ha (2003) describe a technique for generating foils for multiple-choice questions by searching a corpus for semantically similar noun phrases to the answer phrase, and Mitkov et al. (2006) demonstrated that automatic generation and manual correction of questions can be more time-efficient than manual authoring alone. Much of the prior QG research has evaluated systems in specific domains (e.g., introductory linguistics, English as a Second Language), and thus we do not attempt empirical comparisons. Also, prior QG systems have modeled their transformations from source text to questions with complex rules for specific question types. We note that similar patterns might be included in our framework as features for ranking, allowing their utility to be learned from data.

A few pieces of previous research have approached the topic of automatic QG directly. Wolfe (1977) describes early work on automatic QG from text for educational purposes. Mitkov and Ha (2003) developed a system that used rules for creating questions from shallow parses of specific types of sentences (e.g., a rule for creating a question *What do/does/did the S V?* from a sentence with SVO order). Their system ignores constraints on WH-movement and also requires writing entirely new rules for new sentence types. They also describe a technique for generating foils for multiple-choice questions by searching a corpus for semantically similar noun phrases to the answer phrase.

Mitkov et al. (2006) demonstrated that automatic generation and manual correction of questions can be more time efficient than manual authoring alone, which is particularly relevant given that our current system would require vetting by humans, based on the evaluation results.

Kunichika et al. (2004) describe a system for generating questions based on syntactic and semantic analyses which are derived using Definite Clause Grammar (Pereira and Warren, 1986).

Gates (2008) describes a QG system which uses phrase structure parses and the `Tsurgeon` tree manipulation language, very similar to our work. However, as in the work by Mitkov et al., she does not explicitly address constraints on WH-movement and relies on complex rules to transform sentences matching very specific patterns.

There is extensive literature about reading comprehension and the use of technology in schools. We refer the reader to the National Reading Panel's report (National Institute of Child Health and Human Development, 2000) as a useful starting point.

6 Further Research and Development

Our evaluations showed that our current system has far from solved the challenging problem of QG. However, by extending and improving upon our current system, we can progress toward that goal. Our system factors the QG process into multiple stages (derivation of new sentences, transformation into questions, and ranking), enabling more or less independent development of particular stages. Further, the rule-based question transducer in stage 2 is also factored into multiple steps such as subject-auxiliary inversion. To a large extent, the rules for these steps can be modified independently as well. In particular, the rules for marking phrases which are unmovable due to WH-movement constraints operate in parallel, making it straightforward to add or adjust rules to better account for movement constraints.

In order to ensure that changes to specific rules do not break other components, we implemented a suite of unit tests. Each test ensures that a single feature of the system operates as expected. If modification of a particular rule causes another component to fail, the unit test for the failing component will quickly alert the developer of the problem. We intend to extend the unit tests by examining the data from our evaluation. We can then safely modify and extend the rule set to improve coverage and accuracy.

7 Conclusion

We presented a general, modular, three-stage framework for automatic comprehension question generation: (1) extract and derive declarative sentences from a source text; (2) transduce declarative sentences into questions using declarative, general-purpose rules; and (3) statistically rank

the output of overgenerating stages 1 and 2 for acceptability. Incorporation of new NLP components (e.g., paraphrase models) into this framework and improvement of existing ones (e.g., parsing) are expected to benefit this application. A manual evaluation shows that our implementation achieves 43.3% precision-at-10, generating approximately 6.8 acceptable questions per 250 words of source text.

Acknowledgments

References

I. L. Beck, M. G. McKeown, R. L. Hamilton, and L. Kucan. 1997. *Questioning the Author: An approach for enhancing student engagement with text*. International Reading Association, Delaware.

D. M. Bikel, R. Schwartz, and R. M. Weischedel. 1999. An algorithm that learns what's in a name. *Machine Learning*, 34(1-3).

J. Brown, G. Frishkoff, and M. Eskenazi. 2005. Automatic question generation for vocabulary assessment. In *Proc. of HLT/EMNLP*.

N. Chomsky. 1973. Conditions on transformations. *A Festschrift for Morris Halle*, pages 232–285.

M. Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

M. Collins. 2000. Discriminative reranking for natural language parsing. In *Proc. of ICML*.

H. T. Dang, D. Kelly, and J. Lin. 2008. Overview of the TREC 2007 question answering track. In *Proc. of TREC*.

W. B. Dolan and C. Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proc. of IWP*.

B. Dorr and D. Zajic. 2003. Hedge Trimmer: A parse-and-trim approach to headline generation. In *Proc. of Workshop on Automatic Summarization*.

A. Echihabi and D. Marcu. 2003. A noisy-channel approach to question answering. In *Proc. of ACL*.

J. R. Finkel, T. Grenager, and C. Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proc. of ACL*.

D. M. Gates. 2008. Generating reading comprehension look-back strategy questions from expository texts. Master's thesis, Carnegie Mellon University.

O. Glickman, I. Dagan, and M. Koppel. 2005. A probabilistic classification approach for lexical textual entailment. In *Proc. of AAAI*.

A. Goldberg. 2006. *Constructions at Work: The Nature of Generalization in Language*. Oxford University Press, New York.

A. C. Graesser, P. Chipman, B. C. Haynes, and A. Olney. 2005. Autotutor: an intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education*, 48(4):612–618.

2005. *Experiments with Interactive Question-Answering*.

S. M. Harabagiu, S. J. Maiorano, and M. A. Pasca. 2002. Open-domain question answering techniques. *Natural Language Engineering*, 1:1–38.

E. Hovy, U. Hermjakob, and C. Lin. 2001. The use of external knowledge in factoid QA. In *Proc. of TREC*, pages 644–652.

D. Klein and C. D. Manning. 2003. Fast exact inference with a factored model for natural language parsing. In *Advances in NIPS 15*.

H. Kunichika, T. Katayama, T. Hirashima, and A. Takeuchi. 2004. Automated question generation methods for intelligent English learning systems and its evaluation. In *Proc. of ICCE*.

J. R. Landis and G. G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33:159–174.

I. Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *Proc. of ACL*.

W. G. Lehnert. 1978. *The process of question-answering*. Erlbaum, Hillsdale, NJ.

R. Levy and G. Andrew. 2006. Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In *Proc. of LREC*.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19.

G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. 1990. WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4).

R. Mitkov and L. A. Ha. 2003. Computer-aided generation of multiple-choice tests. In *Proc. of the HLT-NAACL 03 workshop on Building educational applications using natural language processing*.

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

J. Mostow, J. Beck, J. Bey, A. Cuneo, J. Sison, B. Tobin, and J. Valeri. 2004. Using automated questions to assess reading comprehension, vocabulary, and effects of tutorial interventions. *Technology, Instruction, Cognition and Learning*, 2:97–134.

National Institute of Child Health and Human Development. 2000. *Report of the National Reading Panel. Teaching children to read: An evidence-based assessment of the scientific research literature on reading and its implications for reading instruction (NIH Publication No. 00-4769)*. U.S. Government Printing Office, Washington, DC.

F. Pereira and D. Warren. 1986. Definite clause grammars for language analysis. *Readings in natural language processing*, pages 101–124.

D. Ravichandran and E. Hovy. 2001. Learning surface text patterns for a question answering system. In *Proc. of ACL*.

E. Reiter and R. Dale. 1997. Building applied natural language generation systems. *Nat. Lang. Eng.*, 3(1):57–87.

J. R. Ross. 1967. *Constraints on Variables in Syntax*. Phd dissertation, MIT, Cambridge, MA.

R. C. Schank. 1986. *Explanation patterns: Understanding mechanically and creatively*. Erlbaum, Hillsdale, NJ.

K. Toutanova, C. Brockett, M. Gamon, J. Jagarlamudi, H. Suzuki, and L. Vanderwende. 2007. The PYTHY summarization system: Microsoft research at duc 2007. In *Proc. of DUC*.

M. A. Walker, O. Rambow, and M. Rogati. 2001. Spot: a trainable sentence planner. In *Proc. of NAACL*.

M. Wang, N. A. Smith, and T. Mitamura. 2007. What is the Jeopardy model? a quasi-synchronous grammar for QA. In *Proc. of EMNLP-CoNLL*.

J. Wolfe. 1977. Automatic question generation from text - an aid to independent study. In *Proc. of ACM SIGCSE-SIGCUE*.