Good Question! Statistical Ranking for Question Generation

Michael Heilman Noah A. Smith
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{mheilman,nasmith}@cs.cmu.edu

Abstract

We address the challenge of automatically generating questions from reading materials for educational practice and assessment. Our approach is to overgenerate questions, then rank them. We use manually written rules to perform a sequence of general purpose syntactic transformations (e.g., subject-auxiliary inversion) to turn declarative sentences into questions. These questions are then ranked by a logistic regression model trained on a small, tailored dataset consisting of labeled output from our system. Experimental results show that ranking nearly doubles the percentage of questions rated as acceptable by annotators, from 27% of all questions to 52% of the top ranked 20% of questions.

1 Introduction

In this paper, we focus on question generation (QG) for the creation of educational materials for reading practice and assessment. Our goal is to 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. Here we restrict our investigation to questions about factual information in texts.

We begin with a motivating example. Consider the following sentence from the Wikipedia article on the history of Los Angeles:[1] *During the Gold Rush years in northern California, Los Angeles became known as the "Queen of the Cow Counties" for its role in supplying beef and other foodstuffs to hungry miners in the north.*

Consider generating the following question from that sentence: *What did Los Angeles become known as the "Queen of the Cow Counties" for?*

We observe that the QG process can be viewed as a two-step process that essentially "factors" the problem into simpler components.[2] Rather than simultaneously trying to remove extraneous information and transform a declarative sentence into an interrogative one, we first transform the input sentence into a simpler sentence such as *Los Angeles become known as the "Queen of the Cow Counties" for its role in supplying beef and other foodstuffs to hungry miners in the north*, which we then can then transform into a more succinct question.

Question transformation involves complex long distance dependencies. For example, in the question about Los Angeles, the word *what* at the beginning of the sentence is a semantic argument of the verb phrase *known as* … at the end of the question. The characteristics of such phenomena are (arguably) difficult to learn from corpora, but they have been studied extensively in linguistics (Ross, 1967; Chomsky, 1973). We take a rule-based approach in order to leverage this linguistic knowledge.

However, since many phenomena pertaining to question generation are not so easily encoded with rules, we include statistical ranking as an integral component. Thus, we employ an overgenerate-and-rank approach, which has been applied successfully in areas such as generation (Walker et al., 2001; Langkilde and Knight, 1998) and syntactic parsing (Collins, 2000). Since large datasets of the appropriate domain, style, and form of questions are not available to train our ranking model, we learn to rank from a relatively small, tailored dataset of human-labeled output from our rule-based system.

The remainder of the paper is organized as fol-

[1] "History of Los Angeles." *Wikipedia*. 2009. Wikimedia Foundation, Inc. Retrieved Nov. 17, 2009 from: http://en.wikipedia.org/wiki/History_of_Los_Angeles.

[2] The motivating example does not exhibit lexical semantic variations such as synonymy. In this work, we do not model complex paraphrasing, but believe that paraphrase generation techniques could be incorporated into our approach.

lows. §2 clarifies connections to prior work and enumerates our contributions. §3 discusses particular terms and conventions we will employ. §4 discusses rule-based question transformation. §5 describes the data used to learn and to evaluate our question ranking model, and §6 then follows with details on the ranking approach itself. We then present and discuss results from an evaluation of ranked question output in §7 and conclude in §8.

2 Connections with Prior Work

The generation of questions by humans has long motivated theoretical work in linguistics (e.g., Ross, 1967), particularly work that portrays questions as transformations of canonical declarative sentences (Chomsky, 1973).

Questions have also been a major topic of study in computational linguistics, but primarily with the goal of *answering* questions (Dang et al., 2008). While much of the question answering research has focused on retrieval or extraction (e.g., Ravichandran and Hovy, 2001; Hovy et al., 2001), models of the transformation from answers to questions have also been developed (Echihabi and Marcu, 2003) with the goal of finding correct answers *given* a question (e.g., in a source-channel framework). Also, Harabagiu et al. (2005) present a system that automatically generates questions from texts to predict which user-generated questions the text might answer. In such work on question answering, question generation models are typically not evaluated for their intrinsic quality, but rather with respect to their utility as an intermediate step in the question answering process.

QG is very different from many natural language generation problems because the input is natural language rather than a formal representation (cf. Reiter and Dale, 1997). It is also different from some other tasks related to generation: unlike machine translation (e.g., Brown et al., 1990), the input and output for QG are in the same language, and their length ratio is often far from one to one; and unlike sentence compression (e.g., Knight and Marcu, 2000), QG may involve substantial changes to words and their ordering, beyond simple removal of words.

Some previous research has directly approached the topic of generating questions for educational purposes (Mitkov and Ha, 2003; Kunichika et al., 2004; Gates, 2008; Rus and Graessar, 2009; Rus and Lester, 2009), but to our knowledge, none has involved statistical models for choosing among output candidates. 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. Existing QG systems model their transformations from source text to questions with many complex rules for specific question types (e.g., a rule for creating a question *Who did the* `Subject Verb`*?* from a sentence with SVO word order and an object referring to a person), rather than with sets of general rules.

This paper's contributions are as follows:

- We apply statistical ranking to the task of *generating* natural language questions. In doing so, we show that question rankings are improved by considering features beyond surface characteristics such as sentence lengths.
- We model 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.
- We incorporate linguistic knowledge to explicitly model well-studied phenomena related to long distance dependencies in WH questions, such as noun phrase island constraints.
- We develop a QG evaluation methodology, including the use of broad-domain corpora.

3 Definitions and Conventions

The term "source sentence" refers to a sentence taken directly from the input document, from which a question will be generated (e.g., *Kenya is located in Africa.*). 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 WH word that replaces an answer phrase (e.g., *Where* in *Where is Kenya located?*).

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

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). The `Tregex` language includes various relational operators based on the primitive relations of immediate dominance (denoted "<") and immediate precedence (denoted "."). `Tsurgeon` adds the ability to *modify* trees by relabeling, deleting, moving, and inserting nodes.

4 Rule-based Overgeneration

Many useful questions can be viewed as lexical, syntactic, or semantic transformations of the declarative sentences in a text. We describe how to model this process in two steps, as proposed in §1.[3]

4.1 Sentence Simplification

In the first step for transforming sentences into questions, each of the sentences from the source text is expanded into a set of derived declarative sentences (which also includes the original sentence) by altering lexical items, syntactic structure, and semantics. Many existing NLP transformations could potentially be exploited in this step, including sentence compression, paraphrase generation, or lexical semantics for word substitution.

In our implementation, a set of transformations derive a simpler form of the source sentence by removing phrase types such as leading conjunctions, sentence-level modifying phrases, and appositives. `Tregex` expressions identify the constituents to move, alter, or delete. Similar transformations have been utilized in previous work on headline generation (Dorr and Zajic, 2003) and summarization (Toutanova et al., 2007).

To enable questions about syntactically embedded content, our implementation also extracts a set of declarative sentences from any finite clauses, relative clauses, appositives, and participial phrases that appear in the source sentence. 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 the next step will produce *What are triggered by computer when prices fall to certain levels?*.

4.2 Question Transformation

In the second step, the declarative sentences derived in step 1 are transformed into sets of questions by a sequence of well-defined syntactic and lexical transformations (subject-auxiliary inversion, WH-movement, etc.). It identifies the answer phrases which may be targets for WH-movement and converts them into question phrases.[4]

In the current implementation, answer phrases can be noun phrases or prepositional phrases, which enables *who*, *what*, *where*, *when*, and *how much* questions. The system could be extended to transform other types of phrases into 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 by applying a series of general-purpose rules. This would allow, for example, the addition of a rule to generate *why* questions that builds 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, and multiple question phrases for each answer phrase. Hence many candidates may result from the transformations.

These rules encode a substantial amount of linguistic knowledge about the long distance dependencies prevalent in questions, which would be challenging to learn from existing corpora of questions and answers consisting typically of only thousands of examples (e.g., Voorhees, 2003).

Specifically, the following sequence of transfor-

[3] See Heilman and Smith (2009) for details on the rule-based component.

[4] We leave the generation of correct answers and distractors to future work.

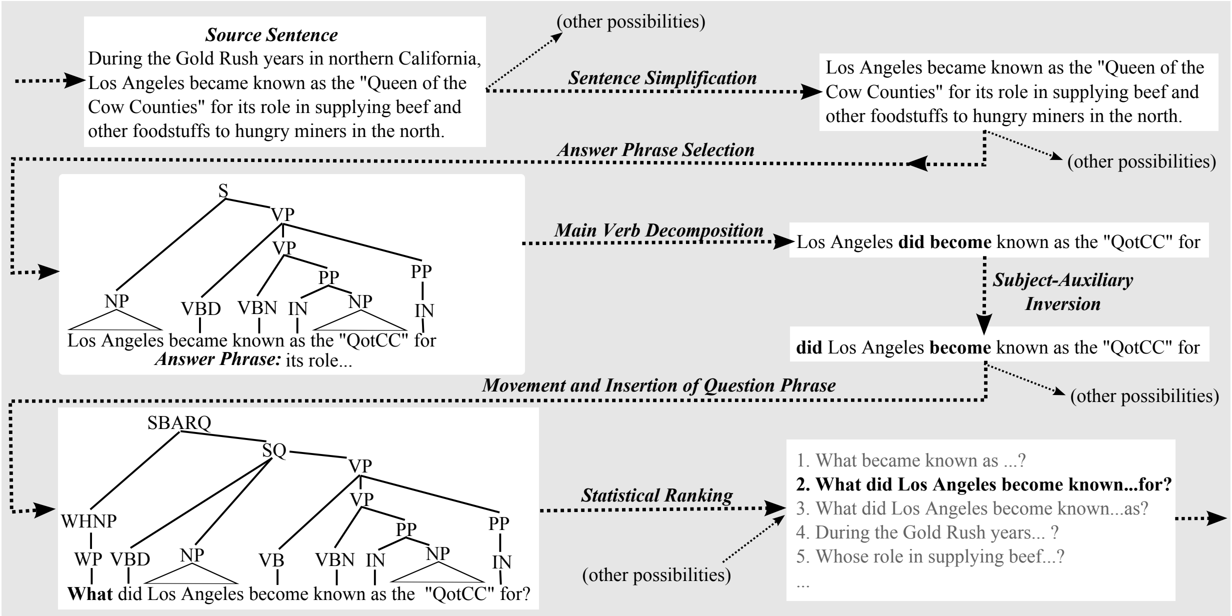
Source Sentence
During the Gold Rush years in northern California, Los Angeles became known as the "Queen of the Cow Counties" for its role in supplying beef and other foodstuffs to hungry miners in the north.
(other possibilities)
Sentence Simplification
Los Angeles became known as the "Queen of the Cow Counties" for its role in supplying beef and other foodstuffs to hungry miners in the north.
(other possibilities)
Answer Phrase Selection
S
VP
VP
PP
PP
NP
VBD
VBN
IN
NP
IN
Los Angeles became known as the "QotCC" for
Answer Phrase: its role...
Main Verb Decomposition
Los Angeles did become known as the "QotCC" for
Subject-Auxiliary Inversion
did Los Angeles become known as the "QotCC" for
Movement and Insertion of Question Phrase
(other possibilities)
SBARQ
SQ
VP
VP
WHNP
PP
PP
WP
VBD
NP
VB
VBN
IN
NP
IN
What did Los Angeles become known as the "QotCC" for?
Statistical Ranking
(other possibilities)
1. What became known as ...?
2. What did Los Angeles become known...for?
3. What did Los Angeles become known...as?
4. During the Gold Rush years... ?
5. Whose role in supplying beef...?
...

Figure 1: An illustration of the sequence of steps for generating questions. For clarity, trees are not shown for all steps. Also, while many questions may be generated from a single source sentence, only one path is shown.

mations is performed, as illustrated in Figure 1: mark phrases that cannot be answer phrases due to constraints on WH movement (§4.2.1, not in figure); select an answer phrase, remove it, and generate possible question phrases for it (§4.2.2); decompose the main verb; invert the subject and auxiliary verb; and insert one of the possible question phrases.

Some of these steps do not apply in all cases. For example, no answer phrases are removed when generating yes-no questions.

4.2.1 Marking Unmovable Phrases

In English, various constraints determine whether phrases can be involved in WH-movement and other phenomena involving long distance dependencies. In a seminal dissertation, Ross (1967) described many of these phenomena. Goldberg (2006) provides a concise summary of them.

For example, noun phrases are “islands” to movement, meaning that constituents dominated by a noun phrase typically cannot undergo WH-movement. Thus, from *John liked the book that I gave him*, we generate *What did John like?* but not **Who did John like the book that gave him?*.

We operationalize this linguistic knowledge to appropriately restrict the set of questions produced. Eight `Tregex` expressions mark phrases that cannot be answer phrases due to WH-movement constraints. For example, the following expression encodes the noun phrase island constraint described above, where `unmv` indicates unmovable noun phrases: `NP << NP=unmv`.

4.2.2 Generating Possible Question Phrases

After marking unmovable phrases, we iteratively remove each possible answer phrase and generate possible question phrases from it. The system annotates the source sentence with a set of entity types taken from the BBN Identifinder Text Suite (Bikel et al., 1999) and then uses these entity labels along with the syntactic structure of a given answer phrase to generate zero or more question phrases, each of which is used to generate a final question. (This step is skipped for yes-no questions.)

5 Rating Questions for Evaluation and Learning to Rank

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, each question is scored according to features of the source sentence, the input sentence, the question, and the transformations used in its generation. The scores are used to rank the questions. This is an example of an "overgenerate-and-rank" strategy (Walker et al., 2001; Langkilde and Knight, 1998).

This section describes the acquisition of a set of rated questions produced by the steps described above. Separate portions of these labeled data will be used to develop a discriminative question ranker (§6), and to evaluate ranked lists of questions (§7).

Fifteen native English-speaking university students rated a set of questions produced from steps 1 and 2, indicating whether each question exhibited any of the deficiencies listed in Table 1.[5] If a question exhibited no deficiencies, raters were asked to label it "acceptable." Annotators were asked to read the text of a newswire or encyclopedia article (§5.1 describes the corpora used), and then rate approximately 100 questions generated from that text. They were asked to consider each question independently, such that similar questions about the same information would receive similar ratings.

For a predefined training set, each question was rated by a single annotator (not the same for each question), leading to a large number of diverse examples. For the test set, each question was rated by three people (again, not the same for each question) to provide a more reliable gold standard. To assign final labels to the test data, a question was labeled as acceptable only if a majority of the three raters rated it as acceptable (i.e., without deficiencies).[6]

An inter-rater agreement of Fleiss's $\kappa = 0.42$ was computed from the test set's acceptability ratings. This value corresponds to "moderate agreement" (Landis and Koch, 1977) and is somewhat lower than for other rating schemes.[7]

5.1 Corpora

The training and test datasets consisted of 2,807 and 428 questions, respectively. The questions were generated from three corpora.

The first corpus was a random sample from the featured articles in the English Wikipedia[8] with 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 on topics of general interest. The training set included 1,328 questions about 12 articles, and the test set included 120 questions about 2 articles from this corpus.

The second corpus 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] The training set included 1,195 questions about 16 articles, and the test set included 118 questions about 2 articles from this corpus.

The third corpus was Section 23 of the *Wall Street Journal* data in the Penn Treebank (Marcus et al., 1993).[10] The training set included 284 questions about 8 articles, and the test set included 190 questions about 2 articles from this corpus.

6 Ranking

We use a discriminative ranker to rank questions, similar to the approach described by Collins (2000) for ranking syntactic parses. Questions are ranked by the predictions of a logistic regression model of question acceptability. Given the question q and source text t, the model defines a binomial distribution $p(R \mid q, t)$, with binary random variable R ranging over a ("acceptable") and u ("unacceptable").

We estimate the parameters by optimizing the regularized log-likelihood of the training data (cf. §5.1) with a variant of Newton's method (le Cessie and

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

[6] The percentages in Table 1 do not add up to 100% for two reasons: first, questions are labeled acceptable in the test set only if the majority of raters labeled them as having no deficiencies, rather than the less strict criterion of requiring no deficiencies to be identified by a majority of raters; second, the categories are not mutually exclusive.

[7] E.g., Dolan and Brockett (2005) and Glickman et al. (2005) report κ values around 0.6 for paraphrase identification and textual entailment, respectively.

[8] The English and Simple English Wikipedia data were downloaded on December 16, 2008 from `http://en.wikipedia.org` and `http://simple.wikipedia.org`, respectively.

[9] The subject matter of the articles in the two Wikipedia corpora was not matched.

[10] In separate experiments with the Penn Treebank, gold-standard parses led to an absolute increase of 15% in the percentage of acceptable questions (Heilman and Smith, 2009).

Question Deficiency	**Description**	**%**
Ungrammatical	The question is not a valid English sentence. (e.g., *In what were nests excavated exposed to the sun?* from . . . *eggs are usually laid . . . , in nests excavated in pockets of earth exposed to the sun.*. This error results from the incorrect attachment by the parser of *exposed to the sun* to the verb phrase headed by *excavated*)	14.0
Does not make sense	The question is grammatical but indecipherable. (e.g., *Who was the investment?*)	20.6
Vague	The question is too vague to know exactly what it is asking about, even after reading the article (e.g., *What do modern cities also have?* from . . . *, but modern cities also have many problems*).	19.6
Obvious answer	The correct answer would be obvious even to someone who has not read the article (e.g., a question where the answer is obviously the subject of the article).	0.9
Missing answer	The answer to the question is not in the article.	1.4
Wrong WH word	The question would be acceptable if the WH phrase were different (e.g., a *what* question with a person's name as the answer).	4.9
Formatting	There are minor formatting errors (e.g., with respect to capitalization, punctuation).	8.9
Other	The question was unacceptable for other reasons.	1.2
None	The question exhibits none of the above deficiencies and is thus acceptable.	27.3

Table 1: Deficiencies a question may exhibit, and the percentages of test set questions labeled with them.

van Houwelingen, 1997). In our experiments, the regularization constant was selected through cross-validation on the training data.

The features used by the ranker can be organized into several groups described in this section. This feature set was developed by an analysis of questions generated from the training set. The numbers of distinct features for each type are denoted in parentheses, with the second number, after the addition symbol, indicating the number of histogram features (explained below) for that type.

Length Features (3 + 24) The set includes integer features for the numbers of tokens in the question, the source sentence, and the answer phrase from which the WH phrase was generated. These numbers of tokens will also be used for computing the histogram features discussed below.

WH Words (9 + 0) The set includes boolean features for the presence of each possible WH word in the question.

Negation (1 + 0) This is a boolean feature for the presence of *not*, *never*, or *no* in the question.

N-Gram Language Model Features (6 + 0) The set includes real valued features for the log likelihoods and length-normalized log likelihoods of the question, the source sentence, and the answer phrase. Separate likelihood features are included for unigram and trigram language models. These language models were estimated from the written portion of the American National Corpus Second Release (Ide and Suderman, 2004), which consists of approximately 20 million tokens, using Kneser and Ney (1995) smoothing.

Grammatical Features (23 + 95) The set includes integer features for the numbers of proper nouns, pronouns, adjectives, adverbs, conjunctions, numbers, noun phrases, prepositional phrases, and subordinate clauses in the phrase structure parse trees for the question and answer phrase. It also includes one integer feature for the number of modifying phrases at the start of the question (e.g., as in *At the end of the Civil War, who led the Union Army?*); three boolean features for whether the main verb is in past, present, or future tense; and one boolean feature for whether the main verb is a form of *be*.

Transformations (8 + 0) The set includes binary features for the possible syntactic transformations (e.g., removal of appositives and parentheticals, choosing the subject of source sentence as the answer phrase).

Vagueness (3 + 15) The set includes integer features for the numbers of noun phrases in the question, source sentence, and answer phrase that are potentially vague. We define this set to include pronouns as well as common nouns that are not specified by a subordinate clause, prepositional phrase, or possessive. In the training data, we observed many vague questions resulting from such noun phrases (e.g., *What is the bridge named for?*).

Histograms In addition to the integer features for lengths, counts of grammatical types, and counts of vague noun phrases, the set includes binary "histogram" features for each length or count. These features indicate whether a count or length exceeds various thresholds: 0, 1, 2, 3, and 4 for counts; 0, 4, 8, 12, 16, 20, 24, and 28 for lengths. We aim to account for potentially non-linear relationships between question quality and these values (e.g., most good questions are neither very long nor very short).

7 Evaluation and Discussion

This section describes the results of experiments to evaluate the quality of generated questions before and after ranking. Results are aggregated across the 3 corpora (§5.1). The evaluation metric we employ is the percentage of test set questions labeled as acceptable. For rankings, our metric is the percentage of the top $N\%$ labeled as acceptable, for various N.

7.1 Results for Unranked Questions

First, we present results for the *unranked* questions produced by the rule-based overgenerator. As shown in Table 1, 27.3% of test set questions were labeled acceptable (i.e., having no deficiencies) by a majority of raters.[11]

The most frequent deficiencies were ungrammaticality (14.0%), vagueness (19.6%), and semantic errors labeled with the "Does not make sense" category (20.6%). Formatting errors (8.9%) were due to both straightforward issues with pre-processing and more challenging issues such as failing to identifying named entities (e.g., *Who was nixon's second vice president?*).

While Table 1 provides data on how often bad questions were generated, a measure of how often good questions were not generated would require knowing the number of *possible* valid questions. Instead, we provide a measure of *productivity*: the system produced an average of 6.0 acceptable questions per 250 words (i.e., the approximate average number of words on a single page in a printed book).

7.2 Configurations and Baselines

For ranking experiments, we present results for the following configurations of features:

All This configuration includes the entire set of features described in §6.

Surface Features This configuration includes only features that can be computed from the surface form of the question, source sentence, and answer phrase—that is, without hidden linguistic structures such as parts of speech or syntactic structures. Specifically, it includes features for lengths, length histograms, WH words, negation, and language model likelihoods.

Question Only This configuration includes all features of questions, but no features involving the source sentence or answer phrase (e.g., it does not include source sentence part of speech counts). It does not include transformation features.

We also present two baselines for comparison:

Random The expectation of the performance if questions were ranked randomly.

Oracle The expected performance if all questions that were labeled acceptable were ranked higher than all questions that were labeled unacceptable.

7.3 Ranking Results

Figure 2 shows that the percentage of questions rated as acceptable generally increases as the set of questions is restricted from the full 428 questions in the test set to only the top ranked questions. While 27.3% of all test set questions were acceptable, 52.3% of the top 20% of ranked questions were acceptable. Thus, the quality of the top fifth was nearly doubled by ranking with all the features.

Ranking with surface features also improved question quality, but to a lesser extent. Thus, unobserved linguistic features such as parts of speech and syntax appear to add value for ranking questions.[12]

The ranker seems to have focused on the "Does not make sense" and "Vague" categories. The percentage of nonsensical questions dropped from 20.6% to 4.7%, and vagueness dropped from 19.6%

[11] 12.1% of test set questions were unanimously acceptable.

[12] Ranking with all features was statistically significantly better ($p < .05$) in terms of the percentage of acceptable questions in the top ranked 20% than ranking with the "question only" or "surface" configurations, or the random baseline, as verified by computing 95% confidence intervals with the BC_a Bootstrap (Efron and Tibshirani, 1993).

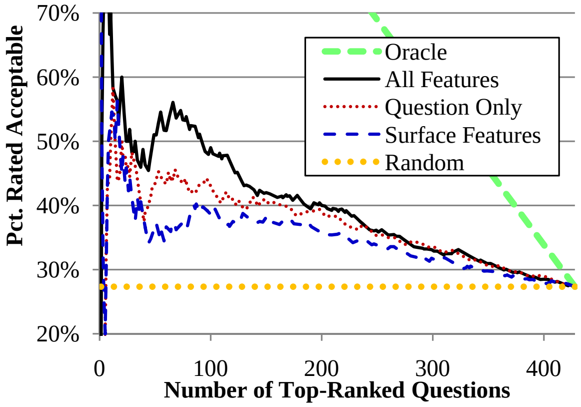

Figure 2: A graph of the percentage of acceptable questions in the top-N questions in the test set, using various rankings, for N varying from 0 to the size of the test set. The percentages become increasingly unstable when restricted to very few questions (e.g., < 50).

Features	#	Top 20%	Top 40%
All	187	52.3	40.8
All – Length	160	52.3	42.1
All – WH	178	50.6	39.8
All – Negation	186	51.7	39.3
All – Lang. Model	181	51.2	39.9
All – Grammatical	69	43.2	38.7
All – Transforms	179	46.5	39.0
All – Vagueness	169	48.3	41.5
All – Histograms	53	49.4	39.8
Surface	43	39.5	37.6
Question Only	91	41.9	39.5
Random	-	27.3	27.3
Oracle	-	100.0	87.3

Table 2: The total numbers of features (#) and the percentages of the top 20% and 40% of ranked test set questions labeled acceptable, for rankers built from variations of the complete set of features ("All"). E.g., "All – WH" is the set of all features *except* WH word features.

to 7.0%, while ungrammaticality dropped from 14.0% to 10.5%, and the other, less prevalent, categories changed very little.[13]

7.4 Ablation Study

Ablation experiments were also conducted to study the effects of removing each of the different types of features. Table 2 presents the percentages of acceptable test set questions in the top 20% and top 40% when they are scored by rankers trained with various feature sets that are defined by removing various feature types from the set of all possible features.

Grammatical features appear to be the most important: removing them from the feature set resulted in a 9.0% absolute drop in acceptability in the top 20% of questions, from 52.3% to 43.3%.

Some of the features did not appear to be particularly helpful, notably the N-gram language model features. We speculate that they might improve results when used with a larger, less noisy training set.

Performance did not drop precipitously upon the removal of any particular feature type, indicating a high amount of shared variance among the features. However, removing several types of features at once led to somewhat larger drops in performance. For example, using only surface features led to a 12.8% drop in acceptability in the top 20%, and using only features of questions led to a 10.4% drop.

8 Conclusion

By ranking the output of rule-based natural language generation system, existing knowledge about WH-movement from linguistics can be leveraged to model the complex transformations and long distance dependencies present in questions. Also, in this overgenerate-and-rank framework, a statistical ranker trained from a small set of annotated questions can capture trends related to question quality that are not easily encoded with rules. In our experiments, we found that ranking approximately doubled the acceptability of the top-ranked questions generated by our approach.

Acknowledgments

We acknowledge partial support from the Institute of Education Sciences, U.S. Department of Education, through Grant R305B040063 to Carnegie Mellon University; and from the National Science Foundation through a Graduate Research Fellowship for the first author and grant IIS-0915187 to the second author. We thank the anonymous reviewers for their comments.

[13] We speculate that improvements in syntactic parsing and entity recognition would reduce the proportion of ungrammatical questions and incorrect WH words, respectively.

References

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

P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2).

N. Chomsky. 1973. Conditions on transformations. *A Festschrift for Morris Halle*.

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*.

B. Efron and R. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman & Hall/CRC.

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.

S. Harabagiu, A. Hickl, J. Lehmann, and D. Moldovan. 2005. Experiments with interactive question-answering. In *Proc. of ACL*.

Michael Heilman and Noah A. Smith. 2009. Question generation via overgenerating transformations and ranking. Technical Report CMU-LTI-09-013, Language Technologies Institute, Carnegie Mellon University.

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

N. Ide and K. Suderman. 2004. The american national corpus first release. In *Proc. of LREC*.

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

R. Kneser and H. Ney. 1995. Improved backing-off for m-gram language modeling. In *Proc. of IEEE Int. Conf. Acoustics, Speech and Signal Processing*.

K. Knight and D. Marcu. 2000. Statistics-based summarization - step one: Sentence compression. In *Proc. of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*.

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.

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

S. le Cessie and J. C. van Houwelingen. 1997. Ridge estimators in logistic regression. *Applied Statistics*, 41.

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.

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).

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).

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

V. Rus and A. Graessar, editors. 2009. *The Question Generation Shared Task and Evaluation Challenge*. `http://www.questiongeneration.org`.

V. Rus and J. Lester, editors. 2009. *Proc. of the 2nd Workshop on Question Generation*. IOS Press.

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*.

E. M. Voorhees. 2004. Overview of the TREC 2003 question answering track. In *Proc. of TREC 2003*.

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