Generating Questions for Reading Comprehension using Coherence Relations

Takshak Desai **Parag Dakle** **Dan I. Moldovan**
Department of Computer Science
The University of Texas at Dallas
Richardson TX
{takshak.desai, paragpravin.dakle, moldovan} @ utdallas.edu

Abstract

We propose a technique for generating complex reading comprehension questions from a discourse that are more useful than factual ones derived from assertions. Our system produces a set of general-level questions using coherence relations. These evaluate comprehension abilities like comprehensive analysis of the text and its structure, correct identification of the author's intent, thorough evaluation of stated arguments; and deduction of the high-level semantic relations that hold between text spans. Experiments performed on the RST-DT corpus allow us to conclude that our system possesses a strong aptitude for generating intricate questions. These questions are capable of effectively assessing student interpretation of text.

1 Introduction

The argument for a strong correlation between question difficulty and student perception comes from Bloom's taxonomy (Bloom et al. (1964)). It is a framework that attempts to categorize question difficulty in accordance with educational goals. The framework has undergone several revisions over time and currently has six levels of perception in the cognitive domain: Remembering, Understanding, Applying, Analyzing, Evaluating and Creating (Anderson et al. (2001)). The goal of a Question Generation (QG) system should be to generate meaningful questions that cater to the higher levels of this hierarchy and are therefore adept at gauging comprehension skills.

The scope of several QG tasks has been severely restricted to restructuring declarative sentences into specific level questions. For example, consider the given text and the questions that follow.

Input: The project under construction will raise Las Vegas' supply of rooms by 20%. Clark county will have 18000 new jobs.

Question 1: What will raise Las Vegas' supply of rooms by 20%?

Question 2: Why will Clark County have 18000 new jobs?

From the perspective of Bloom's Taxonomy, questions like Question 1 cater to the 'Remembering' level of the hierarchy and are not apt for evaluation purposes. Alternatively, questions like Question 2 would be associated with the 'Analyzing' level as these would require the student to draw a connection between the events, 'increase in room supply in Las Vegas' and 'creation of 18000 new jobs in Clark County'. Further, such questions would be more relevant in the context of an entire document or paragraph; and serve as better reading comprehension questions.

This paper describes a generic framework for generating comprehension questions from short edited texts using coherence relations. It is organized as follows: Section 2 elaborates on previously designed QG systems and outlines their limitations. We also discuss Rhetorical Structure Theory (RST), which lays the linguistic foundations for discourse parsing. In Section 3, we explain our model and describe the syntactic transformations and templates applied to text spans for performing QG. In Section 4, we discuss experiments performed on the annotated RST-DT corpus and measure the quality of questions generated by the system. Proposed evaluation criteria address both the grammaticality and complexity of generated questions. We have also compared our system with a baseline to show that our system is able to generate complex questions. Finally, in Section 5, we provide our conclusions and suggest potential avenues for future research.

Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications, pages 1–10
Melbourne, Australia, July 19, 2018.

2 Related Work

2.1 Previous QG systems

Previous research work done in QG has primarily focused on transforming declarations into interrogative sentences, or on using shallow semantic parsers to create factoid questions.

Mitkov and Ha (2003) made use of term extraction and shallow parsing to create questions from simple sentences. Heilman and Smith (2010) suggested a system that over-generates questions from a sentence. Firstly, the sentence is simplified by discarding leading conjunctions, sentence-level modifying phrases, and appositives. It is then transformed into a set of candidate questions by carrying out a sequence of well-defined syntactic and lexical transformations. Then, these questions are evaluated and ranked using a classifier to identify the most suitable one.

Similar approaches have been suggested over time to generate questions, like using a recursive algorithm to explore parse trees of sentences in a top-down fashion (Curto et al. (2012)), creating fill-in-the-blank type questions by analyzing parse trees of sentences and thereby identifying answer phrases (Becker et al. (2012)); or using semantics-based templates (Lindberg et al. (2013); Mazidi and Nielsen (2014)). A common drawback associated with these systems is that they create factoid questions from single sentences and focus on grammatical and/or semantic correctness, not question difficulty.

The generation of complex questions from multiple sentences or paragraphs was explored by Mannem et al. (2010). Discourse connectives such as 'because', 'since' and 'as a result' signal explicit coherence and can be used to generate Why-type questions. Araki et al. (2016) created an event-centric information network where each node represents an event and each edge represents an event-event relation. Using this network, multiple choice questions and a corresponding set of distractor choices are generated. Olney et al. (2012) suggested the use of concept maps to create inter-sentential questions where knowledge in a book chapter is represented as a concept map to generate relevant exam questions. Likewise, Papasalouros et al. (2008) and Stasaski and Hearst (2017) created questions utilizing information-rich ontologies.

Of late, several encoder-decoder models have been used in Machine Translation (Cho et al. (2014)) to automatically learn the transformation rules that enable translation from one language to another. Yin et al. (2015) and Du et al. (2017) argue that similar models can be used to automatically translate narrative sentences into interrogative ones.

2.2 Rhetorical Structure Theory

In an attempt to study the functional organization of information in a discourse, a framework called Rhetorical Structure Theory (RST) was proposed by Thompson and Mann (1987). The framework describes how short texts written in English are structured by defining a set of coherence relations that can exist between text spans. Typically, relations in RST are characterized by three parameters: the nucleus, the satellite and the rhetorical interaction between the nucleus and the satellite. The nucleus is an action; the satellite either describes this action, provides the circumstance in which this action takes place or is a result of the performed action. Notable exceptions are relations such as Contrast, List, etc. which are multinuclear and do not involve satellites.

In order to describe the complete document, these relations are expressed in the form of a discourse graph, an example of which is shown in Figure 1 (O'Donnell, 2000).

We simplify the task of QG by focusing only on the relations given in Table 1. We have condensed some of the relations defined in the RST manual (Thompson and Mann, 1987) and grouped them into new relation types as shown. A complete definition of these relation types can be found in Carlson et al. (2003).

Relation (N,S)	**Obtained from**
Explanation (N,S)	Evidence, Reason, Explanation
Background (N,S)	Background, Circumstance
Cause (N,S)	Cause, Purpose
Result (N,S)	Result, Consequence
Solutionhood (N,S)	Problem-Solution
Condition (N,S)	Condition, Hypothetical
Evaluation (N,S)	Evaluation, Conclusion

Table 1: Set of relations used by our system. Here, N represents the Nucleus and S represents the Satellite

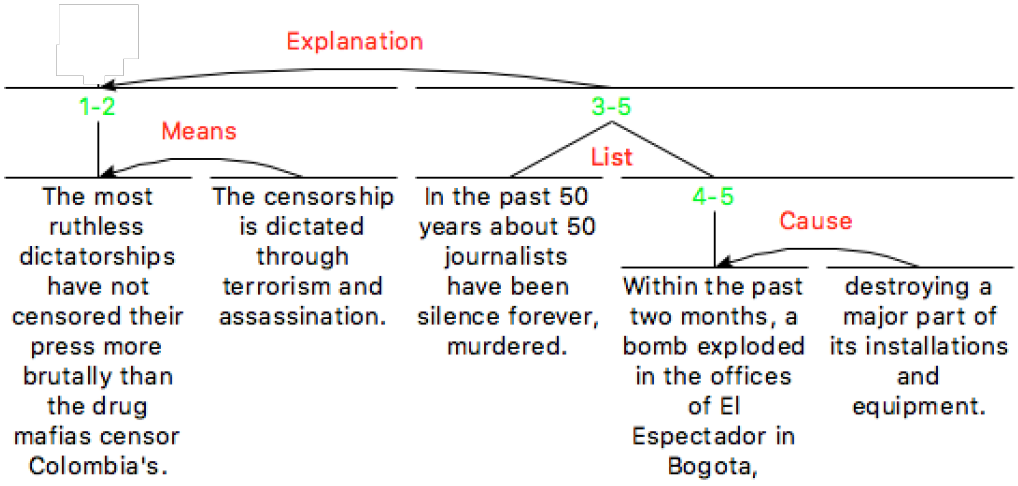

Figure 1: An example of discourse graph for a text sample from the RST-DT corpus

3 Approach

3.1 System Description

The text from which questions are to be generated goes through the pipeline shown in Figure 2. A detailed description of each module/step in the pipeline is described in the subsequent subsections.

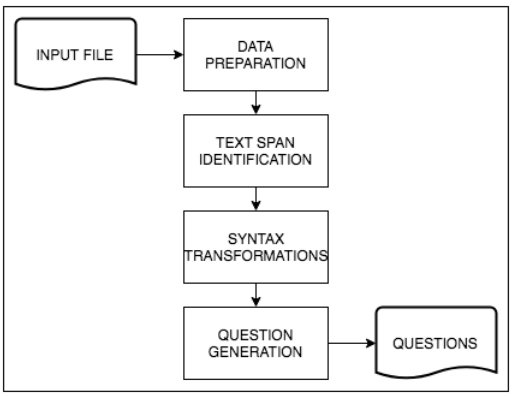

Figure 2: System pipeline

3.1.1 Data Preparation

Here the discourse graph associated with the document is input to the system, which in turn extracts all relevant nucleus-satellite pairs. Each pair is represented as the tuple: Relation (Nucleus, Satellite).

Prior to applying any syntactic transformations on the text spans, we remove all leading and/or trailing conjunctions, adverbs and infinitive phrases from the text span. Further, if the span begins or ends with transition words or phrases like 'As a result' or 'In addition to', we remove them as well.

The inherent nature of discourse makes it difficult to interpret text spans as coherent pockets of information. To facilitate the task of QG, we have ignored text spans containing one word. Further, in several cases, we observe that the questions make more sense if coreference resolution is performed: this task was performed manually by a pair of human annotators who resolved all coreferents by replacing them with the concepts they were referencing. Two types of coreference resolution are considered: event coreference resolution (where coreferents referring to an event are replaced by the corresponding events) and entity coreference resolution (where coreferents referring to entities are replaced by the corresponding entities). Also, to improve the quality of generated questions, annotators replaced some words by their synonyms (Glover et al. (1981); Desai et al. (2016)).

3.1.2 Text-span Identification

We associate each text span with a *Type* depending on its syntactic composition. The assignment of Types to the text spans is independent of the coherence relations that hold between them. Table 2 describes these Types with relevant examples.

3.1.3 Syntax transformations

If the text span is of Type 1 or Type 2, we analyze its parse tree and perform a set of simple surface syntax transformations to convert it into a form suitable for QG. We first use a dependency parser to find the principal verb associated with the span,

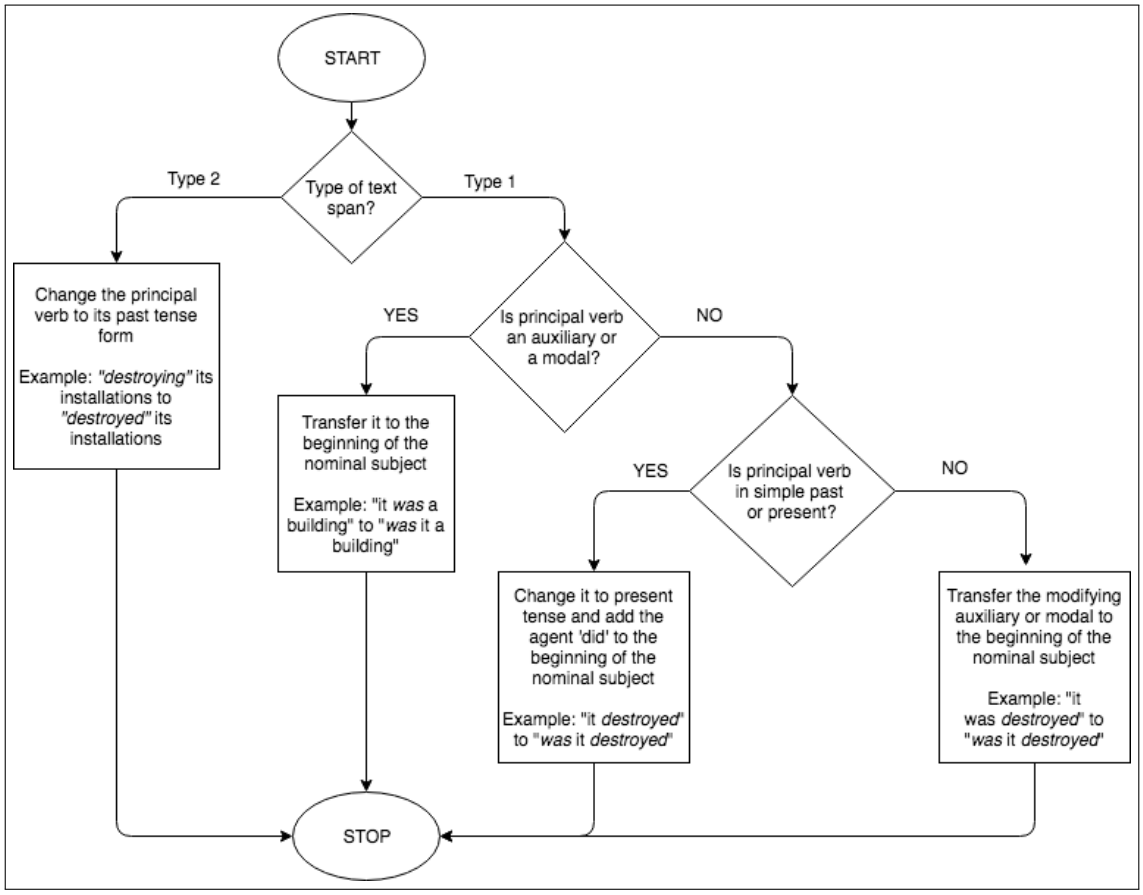

Figure 3: Syntactic transformations applied on text spans. These transformations convert the spans to a form suitable for QG.

Span type	Characteristic of span	Example
Type 0	A group of many sentences	A bomb exploded in the building. It destroyed its installations.
Type 1	One sentence, or a phrase or clause not beginning with a verb, but containing one	The bomb destroyed the building.
Type 2	Phrase or clause beginning with a verb	destroyed the buildings
Type 3	Phrase or clause that does not contain a verb	destruction of the building

Table 2: Text span Types with relevant examples

its part-of-speech tag and the noun or noun phrase it is modifying. Then, according to the obtained information, we apply a set of syntactic transformations to alter the text. Figure 3 describes these transformations as a flowchart.

No syntactic transformations are applied on text spans of Type 0 or Type 3. We directly craft questions from text spans that belong to these Types.

3.1.4 Question Generation

Upon applying the transformations described in Figure 3, we obtain a text form suitable for QG. A template is applied to this text to formulate the final question. Table 3 defines these templates. The design of the chosen templates depends on the relation holding between the spans, without considering the semantics or the meaning of the spans. This makes our system generic and thereby scalable to any domain.

3.2 Example

As an example, consider the same discourse graph from Figure 1. We show how our system will gen-

Relation	Template for type 0	Template for type 1	Template for type 2	Template for type 3
Explanation	[Nucleus]. What evidence can be provided to support this claim?	Why [Nucleus]?	What [Nucleus]?	What caused [Nucleus]?
Background	[Nucleus]. Under what circumstances does this happen?	Under what circumstances [Nucleus]?	What circumstances [Nucleus]?	What circumstances led to [Nucleus]?
Solutionhood	[Nucleus]. What is the solution to this problem?	What is the solution to [Nucleus]?	What solution [Nucleus]?	What is the solution to the problem of [Nucleus]?
Cause	[Satellite]. Explain the reason for this statement.	Why [Satellite]?	What [Satellite] ?	Explain the reason for [Satellite]?
Result	[Nucleus]. Explain the reason for this statement.	Why [Nucleus]?	What [Nucleus] ?	Explain the reason for [Nucleus]?
Condition	[Nucleus]. Under what conditions did this happen ?	Under what conditions [Nucleus]?	What conditions [Nucleus] ?	What conditions led to [Nucleus]?
Evaluation	[Nucleus]. What lets you assess this fact?	What lets you assess [Nucleus]?	What assessment [Nucleus]?	What assessment can be given for [Nucleus]?

Table 3: Templates for Question Generation.

erate questions for a causal relation that has been isolated in Figure 4.

For the given relation, we begin by associating the satellite: "destroying a major part of its installations and equipment" with Type 2. The principal verb 'destroying' is changed to past tense form 'destroyed' and the pronoun 'it' is replaced by the entity it is referencing i.e. 'the offices of El Especatador', to obtain the question stem: 'destroyed a major part of the installations and equipment of the offices of El Especatador'.

We use the template for the cause relation for Type 2 to obtain the question: "What destroyed the installations and equipment of the offices of El Especatador?". Similar examples have also been provided in Table 4.

4 Experimental Results

4.1 Data

For the purpose of experimentation, we used the RST-DT corpus (Carlson et al. (2003)) that contains annotated Wall Street Journal articles. Each article is associated with a discourse graph that describes all the coherence relations that hold between its components. We used these discourse graphs for generating questions. As described in a previous section, we filtered certain relations, and did not consider those relations in which the template is to be applied to text spans containing only one word.

4.2 Implementation

Part-of-Speech tagging and Dependency parsing were performed using Stanford's Part-of-Speech tagger (Toutanova et al. (2003)) and Dependency Parser (Nivre et al. (2016); Bird (2006)) respectively. We used the powerful linguistics library provided by NodeBox (Bleser et al. (2002)) to convert between verb forms. We have used a heavily annotated corpus and made several amendments ourselves, by performing coreference resolution and paraphrasing. This is due to the inability of modern discourse parsers to perform these tasks with high accuracy. While advances have been made in discourse parsing (Rutherford and

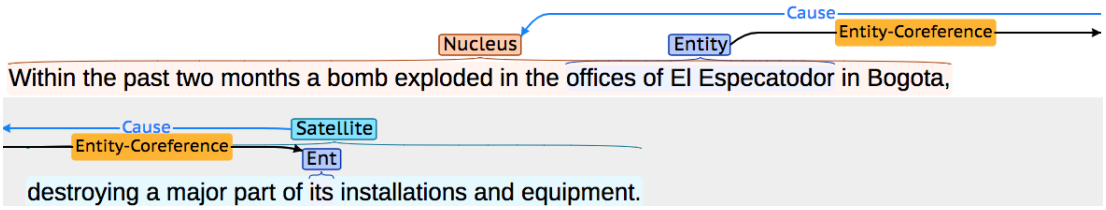

Figure 4: Example of a cause relation from the document

Metric type	Relation	Generated Question	Evaluation
Nature of coherence relation	Nucleus: they are going to be in big trouble with unionists over any Jaguar deal. Satellite: If they try to build it somewhere else in Europe besides the U.K., Relation: Condition	Under what conditions are **General Motors and Ford Motor Co.** going to be in big trouble with unionists over any Jaguar deal?	This is an example of an explicit relation, made apparent through the use of discourse connective 'If' in the satellite
Nature of question	Nucleus: **As a result,** Colombia will earn $500 million less from its coffee this year than last. Satellite: The 27-year old coffee cartel had to be formally dissolved this summer. Relation: Result	Why will Colombia earn $500 million less from its coffee this year than last?	Here, both the question and answer are derived from text spans belonging to different sentences. Thus the score assigned will be 1.
Number of inference steps	Nucleus: **Then,** when it would have been easier to resist them, nothing was done Satellite: and **my** brother was **murdered** by the mafia three years ago Relation: Explanation	Why was **the author's** brother **killed** by the mafia three years ago?	The student should be able to correctly resolve the pronoun 'my' to 'the author' and know that 'killed' is a synonym of 'murdered'. Thus two semantic concepts, paraphrase detection and entity co-reference resolution, are tested here.

Table 4: Examples for metric evaluation

Xue (2014); Li et al. (2014)), such models make several simplifying assumptions about the input. Likewise, coreference resolution (Bengtson and Roth (2008); Wiseman et al. (2016)) is also an uphill task in discourse parsing.

4.3 Evaluation Criteria

To evaluate the quality of generated questions, we used a set of criteria that are defined below. We considered and designed metrics that measure both the correctness and difficulty of the question. All the metrics use a two-point scale: a score of 1 indicates the question successfully passed the metric, a score of 0 indicates otherwise.

- Grammatic correctness of questions: This metric checks whether the question generated is only syntactically correct. We do not take into account the semantics of the question.
- Semantic correctness of questions: We account for the meaning of the generated question and whether it makes sense to the reader.

It is assumed if a question is grammatically incorrect, it is also semantically incorrect.

- Superfluous use of language: Since we are not focusing on shortening sentences or removing redundant data from the text, generated questions may contain information not required by the student to arrive at the answer. Such questions should be refined to make them shorter and sound more fluent or natural.
- Question appropriateness: This metric judges whether the question is posed correctly i.e. we check if the question is not ambivalent and makes complete sense to the reader.
- Nature of coherence relation: Coherence relations are classified into two categories: explicit (the relations that are made apparent through using discourse connectives) and implicit (the relations that require a deep understanding of the text). Questions generated through explicit coherence relations are easier to attempt as compared to the ones generated via implicit coherence relations. We assign a score of 1 to a question generated from an implicit coherence relation and 0 to that generated from an explicit relation.
- Nature of question: We check for the nature of generated question: If both the answer and question are derived from the same sentence, we assign a score of 0, otherwise the score will be 1.
- Number of inference steps (Araki et al. (2016)): To evaluate this metric, we consider three semantic concepts: paraphrase detection, entity co-reference resolution and event co-reference resolution. We consider a score for each concept: 1 if the concept is required and 0 if not. We take the arithmetic mean of these scores to get the average number of inference steps for a question.

4.4 Example

As an example, consider some of the tuples obtained from the RST-DT corpus. Table 4 explains how the generated questions evaluate against some of our criteria.

4.5 Results and Analysis

We generated questions for the entire corpus using our system. For the 385 documents it contains, a total of 3472 questions were generated. Table 5 describes the statistics for the questions generated for each relation type.

Relation type	**Fraction of generated questions**
Explanation	0.282
Background	0.263
Solutionhood	0.014
Cause	0.164
Result	0.156
Condition	0.067
Evaluation	0.054

Table 5: Statistics for Generated Questions

For evaluating our system (represented as QG), we considered the system developed by Heilman and Smith (2010) as a baseline (represented as MH). We sampled 20 questions for each relation type. Note that we did not consider the last four metrics for comparison purposes as these metrics were designed keeping question complexity in mind: MH never addressed this issue and hence such a comparison would be unfair. Table 6 summarizes the results obtained for our system against each relation type. The process was done by two evaluators who are familiar with the evaluation criteria, and are well versed with the corpus and nature of generated questions. The table reports the average scores, considering the evaluation done by each evaluator.

An analysis of the results reveals that many questions are syntactically and semantically well-formed and our results are comparable to that of MH. QG does outperform MH in several cases: however these performance gains are incremental. Issues commonly arose due to errors made by the parser; and the inability of NodeBox to convert between verb forms. Additionally, in some cases, the templates designed were unable to handle all text span Types either due to poor design or because the text span did not follow either definition of the defined Types. For example, some text spans were phrased as questions and some had typographical errors (originally in the text): this led to the generation of unnatural questions. Further, some text spans were arranged in a way such that the main clause appeared after the subordinate clause (For example, the sentence 'If I am hungry, I will eat a cake'): handling such text spans would require us to modify the text such that the subordinate clause

Evaluation criteria	System	R1	R2	R3	R4	R5	R6	R7	Average
Grammatical Correctness	MH	**0.95**	**0.94**	**0.91**	**0.98**	**0.98**	**0.9**	**0.84**	**0.95**
	QG	**0.95**	0.92	**0.91**	**0.98**	0.97	0.87	0.8	0.94
Semantic Correctness	MH	**0.95**	**0.91**	0.97	0.88	**0.94**	**0.88**	**0.8**	**0.93**
	QG	0.93	**0.91**	**0.98**	**0.92**	**0.94**	0.87	**0.8**	0.91
Superfluity of language	MH	**0.84**	**0.81**	0.77	**0.82**	**0.71**	0.9	**0.83**	0.66
	QG	0.81	0.69	**0.78**	**0.82**	0.68	**0.96**	0.8	**0.7**
Question Appropriateness	QG	**0.93**	0.83	**0.95**	**0.75**	**0.78**	**0.87**	0.6	**0.85**
Nature of coherence relation	QG	**0.79**	**0.38**	**1.0**	**0.33**	**0.27**	**0.22**	**0.94**	**0.52**
Nature of Question	QG	**0.71**	**0.37**	**1.0**	**0.24**	**0.24**	**0.4**	**0.88**	**0.45**
Average no. of inference steps	QG	**0.43**	**0.46**	**0.42**	**0.56**	**0.39**	**0.33**	**0.27**	**0.42**

Table 6: Average score for the evaluation criteria. Here R1: Explanation, R2: Background, R3: Solutionhood, R4: Cause, R5: Result, R6: Condition, R7: Evaluation. The average scores for each criterion are indicated in the last column.

follows the main clause (In this example's case, 'I will eat a cake if I am hungry'). However, to the best of our knowledge, there are no known transformations that allow us to achieve this rearrangement.

Table 7 provides some statistics on common error sources that contributed to semantic (and/or grammatical) errors in generated questions.

Source of Error	Percentage of incorrect questions
NodeBox errors	6.7%
Parsing errors	8.3%
Poor template design	13.3%
Incorrect Type Identification	13.3%
Clause rearrangement	57.3%
Other minor errors	1.0%

Table 7: Common error sources: The percentage of incorrect questions is the ratio of incorrect to total questions with semantic/grammatic errors.

Superfluity of language is of concern, as generated questions often contained redundant information. However, identifying redundant information in a question would require a deep understanding of the semantics of the text spans and of the relation that holds between them. Currently, modern discourse parsers are inept at handling this aspect.

The latter four metrics depend heavily on the corpus, and not the designed system. QG, because of its ability to create inter-sentential questions and handle complex coherence relations, was given a moderate to good score by both evaluators. Depending on the text and its relations, these scores may vary. We expect these scores to increase considerably for a corpus containing many implicit relations between text spans that are displaced far apart in the text.

5 Conclusions and future work

We used multiple sources of information, namely a cognitive taxonomy and discourse theory to generate meaningful questions. Our contribution to the task of QG can be thus summarized as:

- As opposed to generating questions from sentences, our system generates questions from entire paragraphs and/or documents.
- Generated questions require the student to write detailed responses that may be as long as a paragraph.
- Designed templates are robust. Unlike previous systems which work on structured inputs such as sentences or events, our system can work around mostly any type of input.
- We have considered both explicit coherence relations that are made apparent through discourse connectives (Taboada (2009)), and implicit relations that are difficult to realize.
- Our system generates inter-sentential questions. To the best of our knowledge, this is

the first work to be proposed that performs this task for a generic document.

There are several avenues for potential research. We have focused only a subset of relations making up the RST-DT corpus. Templates can also be defined for other relations to generate more questions. Further, Reed and Daskalopulu (1998) argue RST can be complemented by defining more relations or relations specific to a particular domain. We also wish to investigate the effectiveness of encoder-decoder models in obtaining questions from Nucleus-Satellite relation pairs. This might eliminate the need for manually performing coreference resolution and/or paraphrasing.

We also wish to investigate other performance metrics that could allow us to measure question complexity and extensibility. Further, we have not addressed the task of ranking questions according to their difficulty or complexity. We wish to come up with a statistical model that analyzes questions and ranks them according to their complexity or classifies them in accordance with the levels making up the hierarchy of Bloom's taxonomy (Thompson et al. (2008)).

References

Lorin W Anderson, David R Krathwohl, P Airasian, K Cruikshank, R Mayer, P Pintrich, James Raths, and M Wittrock. 2001. A taxonomy for learning, teaching and assessing: A revision of blooms taxonomy. *New York. Longman Publishing. Artz, AF, & Armour-Thomas, E.(1992). Development of a cognitive-metacognitive framework for protocol analysis of mathematical problem solving in small groups. Cognition and Instruction* 9(2):137–175.

Jun Araki, Dheeraj Rajagopal, Sreecharan Sankaranarayanan, Susan Holm, Yukari Yamakawa, and Teruko Mitamura. 2016. Generating questions and multiple-choice answers using semantic analysis of texts. In *COLING*. pages 1125–1136.

Lee Becker, Sumit Basu, and Lucy Vanderwende. 2012. Mind the gap: learning to choose gaps for question generation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pages 742–751.

E. Bengtson and D. Roth. 2008. Understanding the value of features for coreference resolution. In *EMNLP*.

Steven Bird. 2006. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*. Association for Computational Linguistics, pages 69–72.

Frederik De Bleser, Tom De Smedt, and Lucas Nijs. 2002. Nodebox version 1.9.5 for mac os x. http://nodebox.net.

Benjamin Samuel Bloom, Committee of College, and University Examiners. 1964. *Taxonomy of educational objectives*, volume 2. Longmans, Green New York.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2003. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Current and new directions in discourse and dialogue*, Springer, pages 85–112.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* .

Sérgio Curto, Ana Cristina Mendes, and Luisa Coheur. 2012. Question generation based on lexico-syntactic patterns learned from the web. *Dialogue & Discourse* 3(2):147–175.

Takshak Desai, Udit Deshmukh, Mihir Gandhi, and Lakshmi Kurup. 2016. A hybrid approach for detection of plagiarism using natural language processing. In *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*. ACM, New York, NY, USA, ICTCS '16, pages 6:1–6:6.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. *arXiv preprint arXiv:1705.00106* .

John A Glover, Barbara S Plake, Barry Roberts, John W Zimmer, and Mark Palmere. 1981. Distinctiveness of encoding: The effects of paraphrasing and drawing inferences on memory from prose. *Journal of Educational Psychology* 73(5):736.

Michael Heilman and Noah A Smith. 2010. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 609–617.

Jiwei Li, Rumeng Li, and Eduard H Hovy. 2014. Recursive deep models for discourse parsing. In *EMNLP*. pages 2061–2069.

David Lindberg, Fred Popowich, John Nesbit, and Phil Winne. 2013. Generating natural language questions to support learning on-line .

Prashanth Mannem, Rashmi Prasad, and Aravind Joshi. 2010. Question generation from paragraphs at upenn: Qgstec system description. In *Proceedings of QG2010: The Third Workshop on Question Generation*. pages 84–91.

Karen Mazidi and Rodney D Nielsen. 2014. Linguistic considerations in automatic question generation. In *ACL (2)*. pages 321–326.

Ruslan Mitkov and Le An Ha. 2003. Computer-aided generation of multiple-choice tests. In *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing-Volume 2*. Association for Computational Linguistics, pages 17–22.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan T McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *LREC*.

Michael O'Donnell. 2000. Rsttool 2.4: a markup tool for rhetorical structure theory. In *Proceedings of the first international conference on Natural language generation-Volume 14*. Association for Computational Linguistics, pages 253–256.

Andrew M Olney, Arthur C Graesser, and Natalie K Person. 2012. Question generation from concept maps. *Dialogue & Discourse* 3(2):75–99.

Andreas Papasalouros, Konstantinos Kanaris, and Konstantinos Kotis. 2008. Automatic generation of multiple choice questions from domain ontologies. In *e-Learning*. Citeseer, pages 427–434.

Chris Reed and Aspassia Daskalopulu. 1998. Modelling contractual arguments. In *PROCEEDINGS OF THE 4TH INTERNATIONAL CONFERENCE ON ARGUMENTATION (ISSA-98). SICSAT*. Citeseer.

Attapol Rutherford and Nianwen Xue. 2014. Discovering implicit discourse relations through brown cluster pair representation and coreference patterns. In *EACL*. volume 645, page 2014.

Katherine Stasaski and Marti A Hearst. 2017. Multiple choice question generation utilizing an ontology. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*. pages 303–312.

Maite Taboada. 2009. Implicit and explicit coherence relations. *Discourse, of course. Amsterdam: John Benjamins* pages 127–140.

Errol Thompson, Andrew Luxton-Reilly, Jacqueline L Whalley, Minjie Hu, and Phil Robbins. 2008. Bloom's taxonomy for cs assessment. In *Proceedings of the tenth conference on Australasian computing education-Volume 78*. Australian Computer Society, Inc., pages 155–161.

Sandra A Thompson and William C Mann. 1987. Rhetorical structure theory. *IPRA Papers in Pragmatics* 1(1):79–105.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, pages 173–180.

Sam Wiseman, Alexander M Rush, and Stuart M Shieber. 2016. Learning global features for coreference resolution. *arXiv preprint arXiv:1604.03035* .

Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li. 2015. Neural generative question answering. *arXiv preprint arXiv:1512.01337* .