Appears in *Proceedings of ACL (the Association for Computational Linguistics)*, 2015.

Deep Questions without Deep Understanding

Igor Labutov
Cornell University
124 Hoy Road
Ithaca, NY
iil4@cornell.edu

Sumit Basu
Microsoft Research
One Microsoft Way
Redmond, WA
sumitb@microsoft.com

Lucy Vanderwende
Microsoft Research
One Microsoft Way
Redmond, WA
lucyv@microsoft.com

Abstract

We develop an approach for generating deep (i.e, high-level) comprehension questions from novel text that bypasses the myriad challenges of creating a full semantic representation. We do this by decomposing the task into an *ontology-crowd-relevance* workflow, consisting of first representing the original text in a low-dimensional ontology, then crowd-sourcing candidate question templates aligned with that space, and finally ranking potentially relevant templates for a novel region of text. If ontological labels are not available, we infer them from the text. We demonstrate the effectiveness of this method on a corpus of articles from Wikipedia alongside human judgments, and find that we can generate relevant deep questions with a precision of over 85% while maintaining a recall of 70%.

1 Introduction

Questions are a fundamental tool for teachers in assessing the understanding of their students. Writing good questions, though, is hard work, and harder still when the questions need to be deep (i.e., high-level) rather than factoid-oriented. These deep questions are the sort of open-ended queries that require deep thinking and recall rather than a rote response, that span significant amounts of content rather than a single sentence. Unsurprisingly, it is these deep questions that have the greatest educational value (Anderson, 1975; Andre, 1979; McMillan, 2001). They are thus a key assessment mechanism for a spectrum of online educational options, from MOOCs to interactive tutoring systems. As such, the problem of automatic question generation has long been of interest to the online education community (Mitkov and Ha, 2003; Schwartz, 2004), both as a means of providing self-assessments directly to students and as a tool to help teachers with question authoring. Much work to date has focused on questions based on a single sentence of the text (Becker et al., 2012; Lindberg et al., 2013; Mazidi and Nielsen, 2014), and the ideal of creating deep, conceptual questions has remained elusive. In this work, we hope to take a significant step towards this challenge by approaching the problem in a somewhat unconventional way.

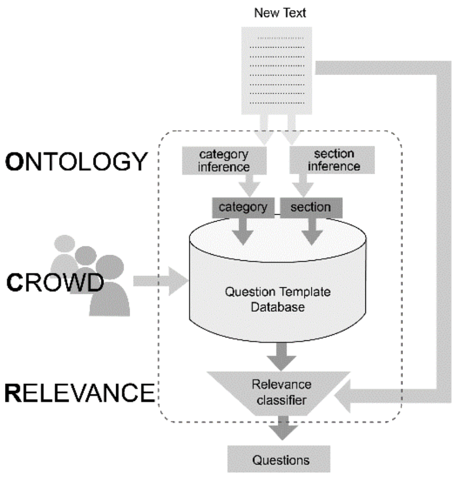


Figure 1: Overview of our ontology-crowd-relevance approach.

While one might expect the natural path to generating deep questions to involve first extracting a semantic representation of the entire text, the state-of-the-art in this area is at too early a stage to achieve such a representation effectively. Rather we take a step back from full understanding, and instead propose an *ontology-crowd-relevance* workflow for generating high-level questions, shown in Figure 1. This involves 1) decomposing a text into a meaningful, intermediate, low-dimensional *ontology*, 2) soliciting high-level templates from the *crowd*, aligned with this intermediate representation, and 3) for a target text segment, retrieving a subset of the collected templates based on its ontological categories and then ranking these questions by estimating the relevance of each to the text at hand.

In this work, we apply the proposed workflow to the Wikipedia corpus. For our ontology, we use a Cartesian product of article categories (derived from Freebase) and article section names (directly from Wikipedia) as the intermediate representation (e.g. category: *Person*, section: *Early life*), henceforth referred to as *category-section pairs*. We use these pairs to prompt our crowd workers to create relevant templates; for instance, (*Person, Early Life)* might lead a worker to generate the question "Who were the key influences on <*Person*> in their childhood?", a good example of the sort of deep question that can't be answered from a single sentence in the article. We also develop classifiers for inferring these categories when explicit or matching labels are not available. Given a database of such *category-section*-specific question templates, we then train a binary classifier that can estimate the relevance of each to a new document. We hypothesize that the resulting ranked questions will be both high-level and relevant, without requiring full machine understanding of the text – in other words, deep questions without deep understanding.

In the sections that follow, we detail the various components of this method and describe the experiments showing their efficacy at generating high-quality questions. We begin by motivating our choice of ontology and demonstrating its coverage properties (Section 3). We then describe our crowdsourcing methodology for soliciting questions and question-article relevance judgments (Section 4), and outline our model for determining the relevance of these questions to new text (Section 5). After this we describe the two datasets that we construct for the evaluation of our approach and present quantitative results (Section 6) as well as examples of our output and an error analysis (Section 7) before concluding (Section 8).

2 Related Work

We consider three aspects of past research in automatic question generation: work that focuses on the grammaticality of natural language question generation, work that focuses on the semantic quality of generated questions, i.e. the "what to ask about" rather than "how to ask it," and finally work that builds a semantic representation of text in order to generate higher-level questions.

Approaches focusing on the grammaticality of question generation date back to the AUTOQUEST system (Wolfe, 1976), which examined the generation of Wh-questions from single sentences. Later systems addressing the same goal include methods that use transformation rules (Mitkov and Ha, 2003), template-based generation (Chen et al., 2009; Curto et al., 2011) and overgenerate-and-rank methods (Heilman and Smith, 2010a). Another approach has been to create fill-in-the-blank questions from single sentences to ensure grammaticality (Agarwal et al. 2011, Becker et al. 2012).

More relevant to our direction is work on the semantic aspect of question generation, which has become a more active research area in the past several years. Several authors (Mazidi and Nielsen 2014; Linberg et al. 2013) generate questions according to the semantic role patterns extracted from the source sentence. Becker et al. (2012) also leverage semantic role labeling within a sentence in a supervised setting. We hope to continue in this direction of semantic focus, but extend the capabilities of question generation to include open-ended questions that go far beyond the scope of a single sentence.

Other work has taken on the challenge of deeper questions by attempting to build a semantic representation of arbitrary text. This has included work using concept maps over keywords (Olney et al. 2012) and minimal recursion semantics (Yao 2010) to reason over concepts in the text. While the work of (Olney et al. 2012) is impressive in its possibilities, the range of the types of questions that can be generated is restricted by a relatively specific set of relations (e.g. Is-A, Part-Of) captured in the ontology of the domain (biology textbook). Mannem et al. (2010) observe as we have that "capturing the exact true meaning of a paragraph is beyond the reach of current NLP systems;" thus, in their system for Shared Task A (for paragraph-level questions (Rus et al. 2010)) they make use of predicate argument structures along with semantic role labeling. However, the generation of these questions is restricted to the first sentence of the paragraph. Though motivated by the same noble impulses of these authors to achieve higher-level questions, our hope is that we can bypass the challenges and constraints of semantic parsing and generate deep questions via a more holistic approach.

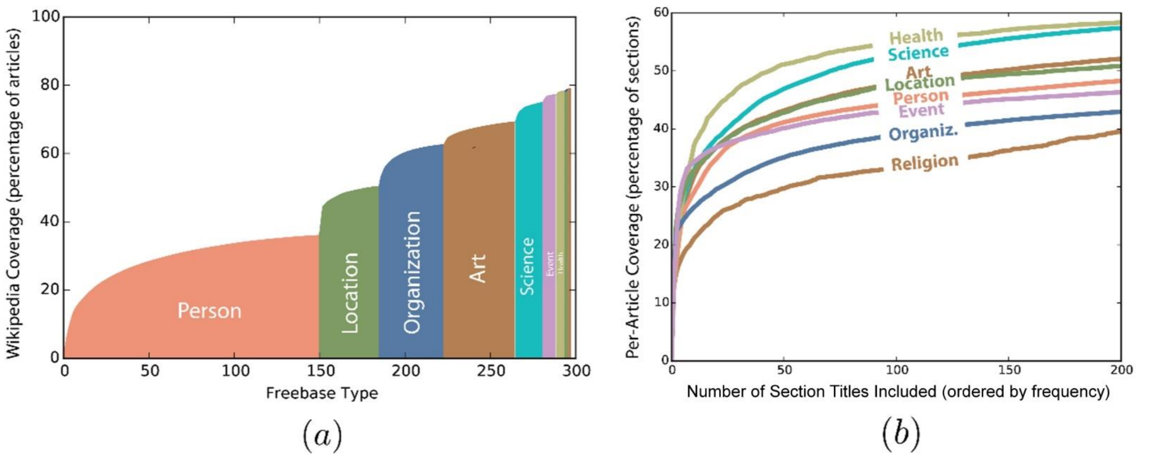


Figure 2: Coverage properties of our *category-section* representation: (a) fraction of Wikipedia articles covered by the top *j* most common Freebase types, grouped by our eight higher-level categories. (b) Average fraction of sections covered per document if only the top *k* most frequent sections are used; each line represents one of our eight categories.

3 An Ontology of Categories and Sections

The key insight of our approach is that we can leverage an easily interpretable (for crowd workers), low-dimensional ontology for text segments in order to crowdsource a set of high-level, reusable templates that generalize well to many documents. The choice of this representation must strike a balance between domain coverage and the crowdsourcing effort required to obtain that coverage. Inasmuch as Wikipedia is deemed to have broad coverage of human knowledge, we can estimate domain coverage by measuring what fraction of that corpus is covered by the proposed representation. In our work, we have developed a *category-section* ontology using annotations from Freebase and Wikipedia (English), and now describe its structure and coverage in detail.

For the high-level *categories*, we make use of the Freebase “notable type” for each Wikipedia article. In contrast to the noisy default Wikipedia categories, the Freebase “notable types” provide a clean high-level encapsulation of the topic or entity discussed in a Wikipedia article. As we wish to maximize coverage, we compute the histogram by type and take the 300 most common ones across Wikipedia. We further merge these into eight broad categories to reduce crowdsourcing effort: *Person, Location, Event, Organization, Art, Science, Health,* and *Religion*. These eight categories cover 78% of Wikipedia articles (see Figure 2a); the mapping between Freebase types and our categories will be made available as part of our corpus (see Section 8).

To achieve greater specificity of questions within the articles, we make use of Wikipedia *sections*, which offer a high-level segmentation of the content. The Cartesian product of our *categories* from above and the most common Wikipedia *section* titles (per category) then yield an interpretable, low-dimensional representation of the article. For instance, the set of *category-section* pairs for an article about *Albert Einstein* contains (*Person, Early_life*), (*Person, Awards*), and (*Person, Political_views*) as well as several others.

For each category, the section titles that occur most frequently represent central themes in articles belonging to that category. We therefore hypothesize that question templates authored for such high-coverage titles are likely to generalize to a large number of articles in that category. Table 1 below shows the four most frequent sections for each of our eight categories.

Person	**Location**	**Organization**	**Art**
Early life Career Pers. life Biography	History Geography Economy Demographics	History Geography Academics Demographics	Plot Reception History Production
Science	**Event**	**Health**	**Religion**
Descript. Taxonomy History Distributn.	Background Aftermath Battle Prelude	Treatment Diagnosis Causes History	Etymology Icongraphy Worship Mythology

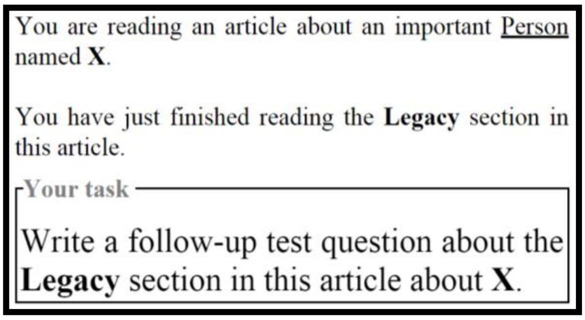
Table 1: Most frequent section titles by category.

As the crowdsourcing effort is directly proportional to the size of the ontology, our goal is to select the smallest set of pairs that will provide sufficient coverage. As with *categories*, the cutoff for the number of *sections* used for each *category* is guided by the trade-off between coverage and crowdsourcing costs. Figure 2b plots the average fraction of an article covered by the top *k* sections from each category. We found that the top 50 sections cover 30% to 55% of the sections of an individual article (on average) across our categories. This implies that by only crowdsourcing question templates for those 50 sections per category, we would be able to ask questions about a third to a half of the sections of any article.

Of course, if we were to limit ourselves to only segments with these labels at runtime, we would completely miss many articles as well as texts outside of Wikipedia. To extend our reach, we also develop the means for category and section *inference* from raw text in Section 5 below, for cases in which ontological labels are either not available or are not contained within our limited set.

4 Crowdsourcing Methodology

We designed a two-stage crowdsourcing pipeline to 1) collect templates targeted to a set of *category-section* pairs and 2) obtain binary relevance judgments for the generated templates in relation to a set of article *segments* (for Wikipedia, these are simply sections) that match in *category-section* labels. We recruit Mechanical Turk workers for both stages of the pipeline, filtering for workers from the United States due to native English proficiency. A total of 307 unique workers participated in the two tasks combined (78 and 229 workers for the generation and ratings tasks respectively).



You are reading an article about an important Person named **X**.

You have just finished reading the **Legacy** section in this article.

Your task

Write a follow-up test question about the **Legacy** section in this article about **X**.

Figure 3: Prompt for the generation task for the *category-section* pair (*Person, Legacy).*

4.1 Question generation task

Following the coverage analysis above, we select the 50 most frequent sections for the top two categories, *Person* and *Location*, yielding 100 *category-section* pairs. As these two *categories* cover nearly 50% of all articles on Wikipedia, we believe that they suffice in demonstrating the effectiveness of the proposed methodology. For each *category-section* pair, we instructed 10 (median) workers to generate a question regarding a hypothetical entity belonging to the target with the prompt in Figure 3. Additional instructions and an interactive tutorial were pre-administered, guiding the workers to formulate appropriately deep questions, i.e. questions that are likely to generalize to many articles, while avoiding factoid questions like “When was X born?”

In total, 995 question templates were added to our question database using this methodology (only 0.5% of all generated questions were exact repeats of existing questions). We confirm in section 4.2 that workers were able to formulate deep, interesting and relevant questions whose answers spanned more than a single sentence and that generalized to many articles using this prompt.

In earlier pilots, we tried an alternative prompt which also presented the text of a specific article segment. In Figure 4, we show the average scope and relevance of questions generated by workers under both prompt conditions. As the figure demonstrates, the alternative prompt showing specific article text resulted in questions that generalized less well (workers’ questions were found to be relevant to fewer articles), likely because the details in the text distracted the workers from thinking broadly about the domain. These questions also had a smaller scope on average, i.e., answers to these questions were contained in shorter spans in the text. The differences in scope and relevance between the two prompt designs were both significant (p-values: 0.006 and 4.5e-11 respectively, via two-sided Welch’s *t*-tests).

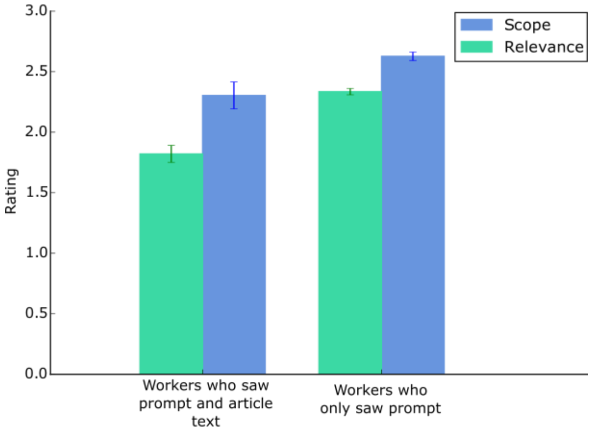


Figure 4: Average relevance and scope of worker-generated questions versus how the workers were prompted.

4.2 Question relevance rating task

For our 100 *category-section* pairs, 4 (median) article segments within reasonable length for a Mechanical Turk task (200-1000 tokens) were drawn at random from the Wikipedia corpus; this resulted in a set of 513 article segments. Each worker was then presented with one of these segments alongside at most 10 questions from the question template database matching in *category-section*; templates were converted into questions by filling in the article-specific entity extracted from the title. Workers were requested to rate each question along three dimensions: *relevance, quality,* and *scope,* as detailed below. Quality and scope ratings were only requested when the worker determined the question to be relevant.

- **Relevance:** *1 (not relevant) – 4 (relevant)*
  Does the article answer the question?
- **Quality:** *1 (poor) – 4 (excellent)*
  Is this question well-written?
- **Scope:** *1 (single-sentence) – 4 (multi-sentence/paragraph)*
  How long is the answer to this question?

A median of 3 raters provided an independent judgment for each question-article pair. The mean relevance, quality and scope ratings across the 995 questions were 2.3 (sd=0.83), 3.5 (sd=.65) and 2.6 (sd=1.0) respectively. Note that the sample sizes for scope and quality were smaller, 774 and 778 respectively, as quality/scope judgments were not gathered for questions deemed irrelevant. We note that 80% of the relevant crowd-sourced questions had a median scope rating larger than 1 sentence, and 23% had a median scope rating of 4, defined as "the answer to this question can be found in many sentences and paragraphs," corresponding to the maximum attainable scope rating. Note that while in this work, we have only used the scope judgments to report summary statistics about the generated questions, in future work these ratings could be used to build a scope classifier to filter out questions targeting short spans of text.

As described in Section 5.2, the relevance judgments are converted to binary relevance ratings for training the relevance classifier (we consider relevance ratings {1, 2} as "not relevant" and {3, 4} as "relevant"). In terms of agreement between raters for these binary relevance labels, we obtained a Fleiss' Kappa of 0.33, indicating fair agreement.

5 Model

There are two key models to our system: the first is for category and section inference of a novel article segment, which allows us to infer the keys to our question database when explicit labels are not available. The second is for question relevance prediction, which lets us decide which question templates from the database's store for that *category-section* actually apply to the text at hand.

5.1 Category/section inference

Both category and section inference were cast as standard text-classification problems. *Category* inference is performed on the whole article, while *section* inference is performed on the individual article segments (i.e., sections). We trained individual logistic regression classifiers for the eight categories and the 50 top section types for each one (a total of 400) using the default L2 regularization parameter in LIBLINEAR (Fan, 2008). For section inference, a total of 736,947 article segments were sampled from Wikipedia (June 2014 snapshot), each belonging to one of the 400 section types and within the same length bounds from Section 4.2 (200-1000 tokens). For category inference, we sampled a total of 86,348 articles with at least 10 sentences and belonging to one of our eight categories.

In both cases, a binary dataset was constructed for a one-against-all evaluation, where the negative instances were sampled randomly from the negative categories or sections (there was an average 17% and 32% positive skew in the section and category datasets, respectively). Basic tf-idf features (using a vocabulary of 200,000 after eliminating stopwords) were used in both text classification tasks. Applying the category/section inference to held-out portions of the dataset (30% for each category/section) resulted in balanced accuracies of 83%/95% respectively, which gave us confidence in the inference. Keep in mind that this is not a strict bound on our question generation performance, since the inferred category/section, while not matching the label perfectly, could still be sufficiently close to produce relevant questions (for instance, we could misrecognize "Childhood" as "Early Life"). We explore the ramifications of this in our end-to-end experiments in Section 6.

5.2 Relevance Classification

We also cast the problem of question/article relevance prediction as one of binary classification, where we map a question-article pair to a relevance score; as such our features had to combine aspects of both the question and the article. Our core approach was to use a vector of the component-wise Euclidean distances between individual features of the question and article segment, i.e., the i^{th} feature vector component f_i is given by $f_i = (q_i - a_i)^2$, where q_i and a_i are the components of the question and article feature vectors. For the feature representation, we utilized a concatenation of continuous embedding features: 300 features from a Word2Vec embedding (Mikolov, 2013) and 200,000 tfidf features (as with category/section classification above).

As question templates are typically short, though, we found that this representation alone performed poorly. As a result, we augmented the vector by concatenating additional distance features between the target article segment and one specific instance of an entire article for which the question applied. This augmenting article was selected at random from all those for which the template was judged to be relevant. The resulting feature vector was thus doubled in length, where the first k distances were between the question template and the target segment, and the next k were between the augmenting article and the target segment. Note that the augmenting article segments were removed from the training/test sets.

To train this classifier, we assumed that we would be able to acquire at least n positive relevance labels for each question template, i.e., n article segments judged to be relevant to each template for inclusion in the training set. We explore the effect of increasing values of n, from 0 (where no relevance labels are available) to 3 (referred to as conditions T0..T3 in Figure 5). We then trained and evaluated the relevance classifier, a single logistic regression model using LIBLINEAR with default L2 regularization, using 10-fold cross-validation on DATASET I (see Section 6).

Figure 5 depicts a series of ROC curves summarizing the performance of our template relevance classifier on unseen article segments. As expected, we see increasing performance with increasing n. However, the benefit drops off after 3 instances (i.e., T4 is only marginally better than T3). While the character of the curves is modest, keep in mind we are already filtering questions by retrieving them from the database for the inferred category-section (which by itself gives us a precision of .74 – see green bars in Figure 6); this ROC represents the “lift” achieved by further filtering the questions with our relevance classifier, resulting in far higher precision (.85 to .95 – see blue bars in Figure 6).

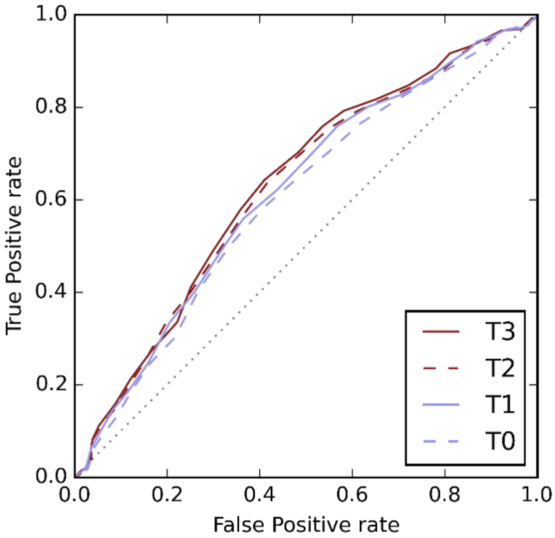


Figure 5: ROC curves for the task of question-to-article relevance prediction. Tn means that n positively labeled article segments were available for each question template during training.

6 Experiments and Results

In this section, we describe the datasets used for training the relevance classifier in Section 5.2 (DATASET I) as well as for end-to-end performance on unlabeled text segments (DATASET II). We then evaluate the performance on this second dataset under three settings: first, when the category and section are known, second, when those labels are unavailable, and third, when neither the labels nor the relevance classifier are available.

6.1 DATASET I: for the Relevance Classifier

The first dataset (DATASET I) was intended for training and evaluating the relevance classifier, and for this we assumed the category and section labels were known. As such, judgments were collected only for questions templates authored for a given article’s actual category and section labels. After filtering out annotations from unreliable workers (based on their pre-test results) as well as those with inter-annotator agreement below 60%, we were left with a set of 995 rated questions, spanning across two categories (*Person* and *Location*) and 50 sections per category (100 *category-section* pairs total). This corresponded to a total of 4439 relevance tuples (*label*, *question*, *article*) where *label* is a binary relevance rating aggregated via majority vote across multiple raters. The relevance labels were skewed towards the positive (relevant) class with 63% relevant instances.

This is of course a mostly unrealistic data setting for applications of question generation (known category and section labels), but greatly useful in developing and evaluating the relevance classifier; we thus used this dataset only for that purpose (see Section 5.2 and Figure 5).

6.2 DATASET II: for End-to-End Evaluation

For an end-to-end evaluation we need to examine situations where the category and section labels are not available and we must rely on inference instead. As this is the more typical use case for our method, it is critical to understand how the performance will be affected. For DATASET II, then, we first sampled articles from the Wikipedia corpus at random (satisfying the constraints described in Section 3) and then performed category and section inference on the article segments. The category c with the highest posterior probability was chosen as the inferred category, while all section types s_i with a posterior probability greater than 0.6 were considered as sources for templates. Only articles whose inferred category was *Person* or *Location* were considered, but given the noise in inference there was no guarantee that the true labels were of these categories. We continued this process until we retrieved a total of 12 articles. For each article segment in these 12, we drew a random subset of at most 20 question templates from our database matching the inferred category and section(s), then ordered them by their estimated relevance for presentation to judges.

We then solicited an additional 62 Mechanical Turk workers to a rating task set up according to the same protocol as for DATASET I. After aggregation and filtering in the same way, the second dataset contained a total 256 (*label, question, article)* relevance tuples, skewed towards the positive class with 72% relevant instances.

6.3 Information Retrieval–based Evaluation

As our end-to-end task is framed as the retrieval of a set of relevant questions for a given article segment, we can measure performance in terms of an information retrieval-based metric. Consider a user who supplies an article segment (the “query” in IR terms) for which she wants to generate a quiz: the system then presents a ranked list of retrieved questions, ordered according to their estimated relevance to the article. As she makes her way down this ranked list of questions, adding a question at a time to the quiz (set Q), the behavior of the precision and recall (with respect to relevance to the article segment) of the questions in Q, summarizes the performance of the retrieval system (i.e. the Precision-Recall (PR) curve (Manning, 2008)). We summarize the performance of our system by averaging the individual article segments’ PR curves (linearly interpolated) from DATASET II, and present the average precision over bins of recall values in Figure 6. We consider the following experimental conditions:

- **Known category/section, using relevance classifier (red):** This is the case in which the actual category and section labels of the query article are known, and only the questions that match exactly in category and section are considered for relevance classification (i.e. added to Q if found relevant by the classifier). Recall is computed with respect to the total number of relevant questions in DATASET II, including those corresponding to sections different from the section label of the article.
- **Inferred category/section, using relevance classifier (blue):** This is the expected use case, where the category/section labels are not known. Questions matching in category and section(s) to the inferred category and section of each article are considered and ranked in Q by their score from the relevance classifier. Recall is computed with respect to the total number of relevant questions in DATASET II.
- **Inferred category/section, ignoring relevance classifier (green):** This is a baseline where we only use category/section inference and then retrieve questions from the database without filtering: all questions that match in inferred category and section(s) of the article are added to Q in a random ranking order, without performing relevance classification.

As we examine Figure 6, it is important to point out a subtlety in our choice to calculate recall of the **known category/section** condition (red bars) with respect to the set of *all* relevant questions, including those that are matched to sections different from the original (labeled) sections. While this condition by construction does not have access to questions of any other section, the resulting limitation in recall underlines the importance of performing section inference: without inference, we achieve a recall of no greater than 0.4.

As we had hypothesized, while the labels of the sections play an instrumental role in instructing the crowd to generate relevant questions, the resulting questions often tend to be relevant to content found under different but semantically related sections as well. Leveraging the available questions of these related sections (by performing inference) boosts recall at the expense of only a small degree of precision (blue bars). If we forgo relevance classification entirely, we get a constant precision of 0.74 (green bars) as mentioned in

Section 5.2; it is clear that the relevance classifier results in a significant advantage.

While there is a slight drop in precision when using inference, this is at least partly due to the constraints that were imposed during data-collection and relevance classifier training, i.e., all pairs of articles and questions belonged to the same category and section. While this constraint made the crowdsourcing methodology proposed in this work tractable, it also prevented the inclusion of training examples for sections that could potentially be inferred at test time. One possible approach to remedy this would be sample from article segments that are similar in text (in terms of our distance metric) as opposed to only segments exactly matching in category and section.

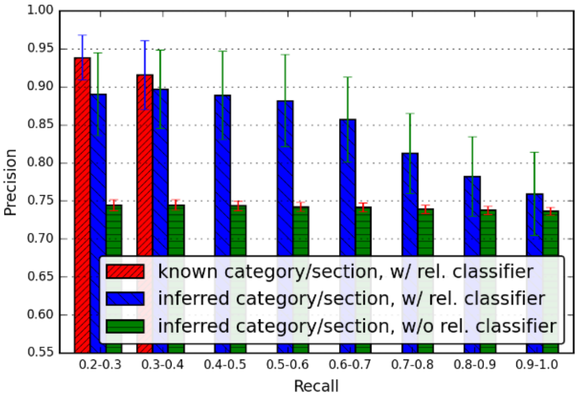


Figure 6: Precision-recall results for the end-to-end experiment, grouped in bins of recall ranges.

7 Examples and Error Analysis

In Table 2 we show a set of sample retrieved questions and the corresponding correctness of the relevance classifier's decision with respect to the judgment labels; examining the errors yields some interesting insights. Consider the false positive example shown in row 8, where the category correctly inferred as *Location*, but section title was inferred as *Transportation* instead of *Services*. This mismatch resulted in the following template authored for (*Location*, *Transportation*) being retrieved: "*What geographic factors influence the preferred transport methods in <entity>?*" To the relevance classifier, this particular template (containing the word "transport") appears to be relevant on the surface level to the text of an article segment about schedules (*Services)* at a railway station. However, as this template never appeared to judges in the context of a *Services* segment – a section that differs considerably in theme from the inferred section (*Transportation)* – the relevance classifier unsurprisingly makes the wrong call.

True section	Inferred section	Result	Generated Question
Honours	Later Life	TP	What accomplishments characterized the later career of Colin Cowdrey?
Acting Career	Television	TP	How did Corbin Bernstein's television career evolve over time?
Route Description	Geography	TP	What are some unique geographic features of Puerto Rico Highway 10?
Athletics	Athletics	TN	How much significance do people of DeMartha Catholic High School place on athletics?
Route Description	Geography	TN	How does the geography of Puerto Rico Highway 10 impact its resources?
Work	Reception	FN	What type of reaction did Thornton Dial receive?
Acting Career	Later Career	FP	What were the most important events in the later career of Corbin Berstein?
Services	Transportation	FP	What geographic factors influence the preferred transport methods in Weymouth Railway Station?
Later Career	Legacy	FP	How has Freddy Mitchell's legacy shaped current events?

Table 2: Examples of retrieved questions. TP, TN, FP, FN stand for true/false positive/negative with respect to the relevance classification.

In considering additional sources of relevance classification errors, recall that we employ a single relevant article segment for the purpose of augmenting a template's feature representation. In the case of the false negative example (row 6 in Table 2), the sensitivity of the classifier to the particular augmenting article used is apparent. Upon inspecting the target article segment (article: *Thornton Dial*, section: *Work*), and the augmenting article segment (article: *Syed Masood*, section: *Reception*), it's clear that the inferred section *Reception* is a reasonable title for the *Work* section of the article on *Thornton Dial*, making the question "What type of reaction did Thornton Dial receive?" a relevant question to the target article (as reflected in the human judgment). However, although both segments generally talk about "reception," the language across the two segments is distinct: the critical reception of Thornton Dial the visual artist is described in a different way from the reception of Syed Masood the actor, resulting in little overlap in surface text, and as a result the relevance classifier falsely rejects the question.

Reasonable substitutions for inferred sections can also lead to false positives, as in row 9, for the article *Freddy Mitchell*. In this case, while *Legacy* (the inferred section) is a believable substitute for the true label of *Later Career*, in this case the article segment did not discuss his legacy. However, there was a good match between the augmenting article for this template and the section. We hypothesize that in both this and the previous examples a broader sample of augmenting article segments for each category/section is likely to be effective at mitigating these types of errors.

8 Conclusion

We have presented an approach for generating relevant, deep questions that are broad in scope and apply to a wide range of documents, all without constructing a detailed semantic representation of the text. Our three primary contributions are 1) our insight that a low-dimensional ontological document representation can be used as an intermediary for retrieving and generalizing high-level question templates to new documents, 2) an efficient crowdsourcing scheme for soliciting such templates and relevance judgments (of templates to article) from the crowd in order to train a relevance classification model, and 3) using category/section inference and relevance prediction to retrieve and rank relevant deep questions for new text segments. Note that the approach and workflow presented here constitute a general framework that could potentially be useful in other language generation applications. For example, a similar setup could be used for high-level summarization, where question templates would be replaced with “summary snippets.”

Finally, to encourage the community to further explore this approach as well as to compare it with others, we are releasing all of our data (category mappings, generated templates, and relevance judgments) at http://research.microsoft.com/~sumitb/questiongeneration .

References

Manish Agarwal, Rakshit Shah, and Prashanth Mannem. 2011. Automatic Question Generation Using Discourse Cues. In *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications*.

Richard C. Anderson and W. Barry Biddle. 1975. On Asking People Questions About What they are Reading. *Psychology of Learning and Motivation.* 9:90-132.

Thomas Andre. 1979. Does Answering Higher-level Questions while Reading Facilitate Productive Learning? *Review of Educational Research* 49(2): 280-318.

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*.

Wei Chen, Gregory Aist, and Jack Mostow. 2009. Generating Questions Automatically from Informational Text. In S. Craig & S. Dicheva (Ed.), *Proceedings of the 2nd Workshop on Question Generation*.

Sérgio Curto, Ana Cristina Mendes, and Luisa Coheur. 2011. Exploring Linguistically-rich Patterns for Question Generation. In *Proceedings of the UCNLG+Eval: Language Generation and Evaluation Workshop*.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research* 9: 1871-1874.

Michael Heilman and Noah Smith. 2010. Good Question! Statistical Ranking for Question Generation. In Proceedings of *NAACL/HLT*.

David Lindberg, Fred Popowich, John Nesbit, and Phil Winne. 2013. Generating Natural Language Questions to Support Learning On-line. In *Proceedings of the 14th European Workshop on Natural Language Generation*.

Prashanth Mannem, Rashmi Prasad, and Aravind Joshi. 2010. Question generation from paragraphs at UPenn: QGSTEC system description. In *Proceedings of the Third Workshop on Question Generation*.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. 2008. *Introduction to Information Retrieval.* Cambridge: Cambridge university press

Karen Mazidi and Rodney D. Nielsen. 2014. Linguistic Considerations in Automatic Question Generation. In *Proceedings of ACL*.

James H. McMillan. 2001. Secondary Teachers' Classroom Assessment and Grading Practices." *Educational Measurement: Issues and Practice* 20(1): 20-32.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of Advances in Neural Information Processing Systems*.

Ruslan Mitkov and Le An Ha. 2003. Computer-Aided Generation of Multiple-Choice Tests. In *Proceed-*

ings of the HLT-NAACL 2003 Workshop on Building Educational Applications Using Natural Language Processing.

Andrew M. Olney, Arthur C. Graesser, and Natalie K. Person. 2012. Question Generation from Concept Maps. *Dialogue & Discourse* 3(2): 75-99.

Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. 2009. Labeled LDA: A Supervised Topic Model for Credit Attribution in Multi-labeled Corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*.

Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Cristian Moldovan. 2010. Overview of The First Question Generation Shared Task Evaluation Challenge. In *Proceedings of the Third Workshop on Question Generation*.

Lee Schwartz, Takako Aikawa, and Michel Pahud. 2004. Dynamic Language Learning Tools. In Proceedings of *STIL/ICALL Symposium on Computer Assisted Learning*.

John H. Wolfe. 1976. Automatic Question Generation from Text - an Aid to Independent Study. In *Proceedings of ACM SIGCSE-SIGCUE Joint Symposium on Computer Science Education*.

Xuchen Yao and Yi Zhang. 2010. Question generation with minimal recursion semantics. In *Proceedings of QG2010: The Third Workshop on Question Generation*.