The Gadfly Project

automatic question generation as a service

Vijay Velagapudi • Andrew Huang • Nikhil Mane • Daniel Griffin • Anand Rajagopal

Advisor: Marti Hearst

Description

Asking questions has inherent value. We want to increase and expand this value with an extensible automatic question generation service to allow interactions with texts that were previously unfeasible. Give our service any block of text and we will automatically return a set of related questions.

Since we are developing an extensible service, the potential use cases are limitless. We decided to work in the area of news publications because there is an inherent discourse surrounding the public's consumption of news media. We think our automatic question generation service can provide value in helping readers understand the articles that they have read, and also perform various functions like comment moderation, quizzing, and trivia applications.

At the foundation of our project is the Gadfly API which is currently powering our custom chat bots (QBot and TriviaBot) living in the I School Slack team, as well as a web interface. The general public can use our service at GadflyProject.com

Table of Contents

Acknowledgements

We would like to thank Marti Hearst for teaching us that, yes, NLP *is* hard, for encouraging us in challenging accepted industry convention, and for reminding us to have fun. We thank Steve Fadden, David Bamman, and Jez Humble for the opportunity to engage with portions of this project in their classes. We also thank Sabhanaz Rashid Diya for her help in our user research. We are grateful for the cooperation of Lance Knobel (co-founder, Berkeleyside), Alison Fu (online managing editor, Daily Cal), Katie Dowd (senior producer, SFGate), Daniel Ha (CEO, Disqus) and Tamara Straus (former Editorial Director, Blum Center, UC Berkeley). We are also indebted to our friends and family and their family and friends who served as our exploratory interviewees, responded to our surveys, participated in our usability studies, and subjected themselves to our experimentation.

Introduction

The Gadfly Project, this automatic question generation service with identified use cases relating to discourse on news, is the end-result of questions we have attempted to answer over the last several months in cycles of research and design. What if Socrates were alive today? Better, what if we could clone many Socrateses?[1] What would he do?

These are the questions that we did not start our project with but wish we had. Instead, recognizing the role of news in our lives we began exploring the discovery and consumption of and discourse around news online. We conducted exploratory interviews with our friends and family, identifying many frustrations.

They highlighted concerns about whether others had actually read what they posted, whether they themselves satisfactorily understood what they shared, frustrations with keeping up with the news, and general horror at the state of commenting on publisher sites.

Problem

“Comments are making the internet worse,” starts the headline of a recent piece where the managing editor of a blog described a turn away from civility in the comments on his site--a turn that convinced his team to get rid of their comment section (Lat, 2016). He went on to list several publishers that have removed their comment sections: Popular Science, Recode, Mic, the Week, Reuters, Bloomberg, and the Daily Beast. This, disabling commenting, is being

[1] "Would there were more Socrateses in the world..." (Allbutt, Thomas Clifford. Notes on the Composition of Scientific Papers. Macmillan, 1905.)

suggested as the only way forward for publishers without the resources of papers such as The New York Times. We recognize the well-documented abuse that occurs in comment sections online and the legal, social, and technical complexities involved in addressing it (Santana, 2014).

Closing comment sections may push more conversation onto social media, where there is also well-documented abusive and harassing behavior (Citron, 2014; Jeong, 2015). As identified in our generative research phase, the quality of social media discussion on news and news articles has room for improvement. Interlocutors cannot always presume that even the person posting or sharing an article has read more than the headline.

While much can be written about the techno-determinism of both the fears of those who decry online comments and of the promises of those providing solutions, we want to upframe here. Is getting rid of comments the world we want? Do we want to relegate all our digital public fora to the private entities like Facebook, Twitter, or Slack? Perhaps not, and perhaps we want entirely different digital public fora. But, we do all know, Cueball is justified in his concern.[2]

The political divide appears to be increasing, with "Democrats and Republicans more ideologically divided than in the past" (Pew, 2016). Not only that, it appears that "partisanship is a political *and* social divide" [emphasis in the original]. Marriage across party lines is reducing (Iyengar and Westwood, 2015). Habits of news consumption has been found to influence political ideology (Pew, 2014). The amount, type, and quality of engagement with and discourse on news matters.

Engagement with and discourse on news may be positively effected through reader interaction with automatically generation of questions from news articles. Questions have inherent value. But their generation is costly - in time and attention. So questions are not applied in every domain. There may be some domains where the value gained from questions may be smaller than the cost of human-generation, but a value nonetheless. We posit that one such domain may be that of discourse on and around news articles.

We do not suppose a tool providing the automatic generation of questions will solve any of these issues. We do propose studying how people may make use of such a tool to address some of these issues.

[2] https://xkcd.com/386/

Research Questions

We began with two primary questions that looked at both the social and technical issues in this space. Research and design work on either necessarily fed into and incorporated feedback from the other.

> Where might automatically-generated questions improve public or social discourse or individual engagement with news?
> How can an automatic question generation service be built to explore this problem area within the scope of this project?

For the first question our initial hypotheses were informed by our exploratory interviewing. We considered use cases for social media sharing but settled first on exploring commenting on news articles. Our hypothesis was that a sort of comment-CAPTCHA service that asked readers automatically generated questions from the article they wished to comment on, before allowing said commenting, may improve the amount, quality, or type of discourse in comment sections. Further into our research and design, for reasons discussed below, we came upon two more use cases: news discovery and trivia. We considered the implementation of two types of bot users — QBot and TriviaBot — on the Slack messaging platform. Slack provided a new distribution opportunity for our use case since we could reach out to users of the I School Slack and get quick feedback from them. QBot exposes a news discovery service while TriviaBot simulates the experience of playing with news based trivia. Both of these bots are powered by The Gadfly API.

Our working hypothesis on our second question was that an implementation using syntactic-level features and rule-based heuristics relying on open source libraries could be built within our budget, team size, theory-acquisition and skills-improvement potential, and the allocated time-frame and our conflicting time-demands.

This report will describe the research we conducted, the design we developed, an evaluation, future work, and conclude with a reassessment of our problem statements.

Research

User Research

We are conducting user research for the project to see if this idea meets a valid pain point both from the readers' and also the newspapers' perspectives. We see ourselves as guiding the team though the different phases of product development. We first conducted generative user interviews with newspaper publishers to discover how they currently use commenting systems for moderation and also their preferences in and methods for influencing discourse around their content.

One constraint from a product development point of view is that the technology being developed relies on using news articles that are available to the public. This allows the team to not worry about issues relating to access or copyright. With this constraint in mind, we leveraged the user research to guide us through the generative, formative, and summative steps to build the best product possible using their extensible automatic question generation technology.

Generative: Interviews (learning about comments usage by publishers)

Interviews were conducted with publishers that are local in the Bay Area. Our assumption was that local newspapers would be more open to the technological culture of Silicon Valley and that openness might lend them to be more willing to partner with us on the proposed comment-CAPTCHA system. We also understood that being able to meet face to face would allow for a more in-depth conversation while being able to read/react to body cues, build trust, and perhaps foster a longer term partnership.

Luckily, most newspaper websites have a directory of employees that can be contacted through email, which became our primary method of reaching out to potential interviewees. We also searched through our social networks for connections, with a few friends being former journalists. In certain cases, where we couldn't locate an email address, we would end up tweeting at the user or leaving our information through a general form on the website. These last two methods were not productive, while emailing resulted in the most responses. We were able to interview three local newspaper publishers (Berkeleyside, SFGate, Daily Cal) and the CEO of a comment moderation provider (Disqus). We also interviewed a company in the education space (CourseHero) to explore other possible use cases for future development. All except one interview was recorded on audio, and all interviews had a notetaker as well. Our interview with the educational company also included engineers from the team since the company had a clear use case for our technology, and we guessed they would have technical

questions for whether we could incorporate our extensible application into their current systems.

Interview Findings

We learned that, in general, newspaper publishers value quantity of comments over quality of comments. Smaller community publications such as Berkeleyside put in extra effort to ensure that they have quality commenting, but it is also due to the fact that they don't receive an overwhelming amount of comments on a regular basis, thus allowing their staff to moderate every comment. In relation to the comment-CAPTCHA idea, the general concerns dealt with increasing the level of friction for readers, and also conveying a sense of condescension. Both are legitimate worries from the publisher's point of view, and helped guide feedback to the product development at an early stage. This allowed for the team to have enough time to explore further options and to pivot to another idea. As for the user research, it solidified a set of questions to be required in our survey, most importantly the question asking whether a reader would rather answer a question or login to leave a comment.

One positive note on the comment-CAPTCHA use case was that all news publishers were curious about the idea and indicated that they would appreciate seeing a demo as we received more feedback and were further along in our technical development. In a larger sense, two of the publishers (Berkeleyside and SF Gate) mentioned that they would be interested in writing a news article about the work we were doing and possibly implement our comment moderation system for that article.

An interview with Daniel Ha, the CEO of Disqus (the comment moderation system used by Berkeleyside and Daily Cal), brought up the theme of engagement with the news. We discussed how questions could be a good way to spur readers to leave a comment. For example, we could ask an open ended question at the end of a news article that would prompt the reader to respond by leaving a comment. This idea was brought back to the team for discussion. The outcome was that generating an open ended question may be too easy and more of a manual solution in which a general set of open ended questions may be applied to any article. The key area of interest that the team wanted to explore was the ability to automatically generate questions that are uniquely related to the text. With this in mind, along with the initial key point of user engagement, the team continued to brainstorm more use cases that could provide value using the automatic question generation technology to potential partners and users.

We decided to pivot most of our design efforts to different use cases, user bots on Slack to test engagement and get feedback on question type and quality, while continuing to investigate questions related to the comment-CAPTCHA use case.

Formative: Survey (understanding people's habits and preferences)

We created our online survey using Qualtrics which is provided free of cost through UC Berkeley. Using Qualtrics allowed us to insert logic into the flow, so that certain responses would allow the user to skips questions that were not applicable. This was important as keeping our survey as short and user-friendly as possible were top priorities. We also used a photo elicitation technique to help the user imagine a scenario which the team wanted to verify was of interest to the general public.

Before sending it out to the public, the team ran through the questions a few times first to test the flows and skip logic of questions. We also looked for questions and answer choices that we thought were badly worded or misguided and leading. When it was deemed ready to make public, we leveraged our social networks and involvement in different Slack teams to gain as much exposure as possible. We highlighted the fact that it should take under 5 minutes to complete, was anonymous, and also emphasized that this was work being conducted by a group of students and friends. This was done in an effort to effectively ask for help across our networks. We let the survey run for 10 days and received 181 responses with an 82% completion rate. A link to the raw survey data is included in the appendix.

Survey Results

The survey's most important finding was that 77% of the respondents would rather answer a question than login or any other method in order to leave a comment. One concern with this statistic though, is that we also found that 87% of the respondents never leave a comment on a weekly basis. This signals to the research team that we need to conduct further interviews with people who regularly leave comments to learn more about their habits and preferences when reading and commenting on the news. However, at the same time, it is enlightening because of the contrast with the existing process of leaving comments that publishers are more comfortable with. With further interviews, we can possibly gather more support for this hypothesis and develop a meaningful use case that can challenge the status quo.

This finding, that 77% of our sample stated they would prefer to answer a question instead of signing into a service before commenting, has encouraged our team to explore this further. It encourages exploration of whether readers prefer to make reading more interactive. Is commenting the only avenue to make a publishing platform user-driven and interactive? Perhaps. Perhaps, there are some fundamental changes to the way people can comment that will make the platform more interactive in real-time and therefore, generate more meaningful conversations.

In terms of reading habits, we learned that over 75% of the survey respondents read a comment every week and gathered general publisher preferences, with many people mentioning the New York Times as having the best comment section.

When asked “Would you login with some form of credentials to leave a comment?”, 66% of the surveyors stated a preference not to login in order to leave a comment. This could be because of an additional step that can lead to opening a new window, the reader’s discomfort with sharing private information, or it being time-consuming in general. In order to understand why readers have such a high preference to not login, which is what existing publishing sites require, we will need take interviews and probe into the reader’s persona, preferences and habits.

At the end of the survey 40 respondents indicated that they would like to interact with our Slack bot and also volunteered their email to help us in further research.

Summative: Usability Testing (assess quality of questions and interactions)

Our usability tests were all conducted in-person. We had ended our survey by asking for interested parties to leave their email if they would like to help us further. We narrowed those email addresses to those that we knew were local to Berkeley in order to expedite the usability tests. Understanding that we had a long list of potential supporters that weren’t part of the I School Slack team, we also created another team so that tests could be done with people that aren’t affiliated with the I School. We used QuickTime to record the screen and audio. We also had a team member present to take notes. One of the usability tests failed on the laptop, so we had to switch computers and continue recording by using a smartphone video camera. Some of the usability tests were conducted with the API engineering team in attendance as well. This allowed for immediate and direct feedback into the prototypes being developed. Apart from the email list from our survey, we also approached heavy Slack users. We wanted people that were extremely comfortable with daily use of the application, since it was hard for us to get the planned tasks completed with those that are unfamiliar with conversing over a messaging platform. We did this by contacting members of the admin private group on the I School slack team. At the time of writing the report we have conducted four usability tests on the initial QBot. Our findings led us to create a separate bot to handle a use case of trivia. We conducted a quick usability test with TriviaBot that quickly uncovered many technical issues that we need to rectify before proceeding with further tests.

Usability Testing Results

Our usability tests quickly showed some immediate flaws that needed to be fixed. The initial prototype only generated gap-fill questions. Testers complained that the questions being asked were too specific and “nitpicky”. We observed them freeze up when conversing with QBot,

since they were not able to guess the answer. There was constant confusion with how to proceed when they did not know the answer, and there was also criticism about the content of the question itself. We had asked the testers to pick an article that they had read to be tested on, so when they were faced with our questions, they quickly saw that they were not representative of the articles they had picked.

When asked about what would alleviate the confusion with the question itself, they requested questions that were more generally related to the article, and they wanted hints, either in the form of multiple-choice questions, or from asking QBot to provide more information upon request. This feedback was immediately relayed to the engineering team since they had already been working on generating multiple-choice questions, but had not explored a more general question generation format at that time.

The feedback also brought up a key struggle throughout this process. The team has been acutely aware that they are working on providing a new service that has been untested before. The most obvious worry with the whole product development process has been focused on a central question: What is the quality of questions that we can generate using algorithms? Although we have worked to incorporate this question into our user research by allowing for feedback mechanisms in interactions with our custom Slack bots, we also have part of the team running different tests to work towards a machine learning method of improving the quality of questions being generated.

This also posed the general challenge of how we went about developing prototypes to test. We were not just taking existing pieces of code and building a website or app to test. The engineering team was writing completely new methods in order to crack the challenge of automatically generating questions. There were plenty of times that we wanted to use the Wizard of Oz technique in the question generation and test that way, but at the foundation of the project, we had to incorporate the quality of questions we were generating. Hence, our usability tests were delayed until the team thought that we had a general level of acceptability in the application.

Experiment

We ran an experiment that was aimed at testing hypotheses to compare between multiple choice, true/false and gap-fill questions. We focused on understanding the differences in user preference and performance between these different types of automatically-generated questions to better allocate the development efforts as well as provide evidence to online publishers around user preferences for different question types in the news domain.

Motivation

The motivation to explore question evaluation stems from the fact that no "standard" evaluation task exists for automatic question generation (Heilman and Smith, 2009). Part of the difficulty of evaluating the effectiveness of question generation is that there are many dimensions upon which a question may or may not be "acceptable", including relevance to various contexts and goals, grammar, and difficulty. While this project is heavily influenced by the academic papers (see NLP Research below), we attempted to make some modifications to the process to better reflect the effectiveness of questions in a setting outside the education domain. The goals of the Gadfly Project are vastly different from the academic focused work on which it is based and, therefore, required a different type of evaluation than previous works. The following subsections explain in more detail some of these differences and how the proposed experiment sought to address them.

The Importance of User Engagement

One of the primary differences from previous work in automatic question generation was the fact that is important for our system was the effect on user engagement. The findings from our interviews conducted with the various stakeholders in the news domain showed us that most publishers were afraid of, or were deeply focused on, the effect questions would have on user engagement. Their primary focus was on whether questions on news articles would be interesting enough to keep users engaged and how they could minimize the potential of a chilling effect on their comment sections.

Experiment Design

We created an experiment on the Qualtrics online survey platform to help us record user preference between the different question types. See section on Articles and Generated Questions in Appendix IId for the questions that went into the survey.

The experiment worked as follows:

1. User was prompted to read a body of text (a news article)
2. User was then presented with six questions (two per page over three pages)[3]
    a. two gap-fill questions[4]
    b. two multiple choice questions
    c. two true/false questions
3. User was prompted to read a second body of text (a news article)

[3] Users were not allowed to see the article during the question answering phase

[4] Identified as fill-in-the-blank to users

4. User was presented with the choice of answering gap-fill, multiple choice, or true/false questions
5. User was asked to explain their reasoning for their question type preference

We were very particular about the choice of news articles to include in this experiment since it could potentially introduce biases such as the differing interest of users in a particular news domain or a user having previously read the article. In order to control for this, we decided to constrain the news articles chosen for our experiment to recent, general topic news collected from the New York Times.[5]

Since each user would only see two articles, we randomized which article each user was given. Therefore, users could have seen four possible combinations of articles 1 & 3, 1 & 4, 2 & 3), and 2 & 4 (see Appendix IId: Table 1).

Experiment Measurements

We measured three different metrics during our experiment, (1) user preference for second article, (2) performance, and (3) time taken to answer question.

User Preference

We decided to treat user preference as a sort of compound metric that was a proxy for whether or not a user found a given question interesting in this context. We briefly explored the idea of asking the user which questions they found most interesting but decided against it because it would be difficult for users to answer that question in any meaningful way. Our assumption was that by asking users which question type they would actually prefer to see again, we would get a more honest answer. Based on earlier research, our hypothesis was that users would have a preference towards MC questions.

Performance

Our hypothesis was that multiple choice questions would be the easiest for users to answer because they did not require users to remember minute details from the article, which would be necessary to answer most gap-fill questions. This we could only test by measuring their performance on those question types. Secondly, we wanted to see if there would be any correlation between user choice of question type for the second article and their performance on the first article.

[5] The following New York Times articles were used in our experiment: Bernie Sanders Campaign Hopes an Endorsement Resonates in New York; The Islamic State of Molenbeek; Egypt Gives Saudi Arabia 2 Islands in a Show of Gratitude; Why Apple's Stand Against the F.B.I. Hurts Its Own Customers

Time to Answer

We also tracked the time it took to answer a question type to attempt to gauge the difficulty of different question types. This measure was used to cross-reference with question accuracy since we hypothesize that accuracy would be inversely correlated with the amount of time taken to answer. We believed that we could use accuracy and time taken as a compound metric of question difficulty.

Experiment Results

Response Statistics

We had 62 people start our experiment and 36 people complete it, which gave us a completion rate of 58%. The counts for each of the possible combinations of articles is listed in the appendix (see Appendix IId: Table 2).

User Preference

Out of the 36 complete responses that our experiment generated, 24 users (66%) preferred to answer multiple choice questions for the second article. Nine users (25%) preferred to answer true/false questions and three users (8%) preferred gap-fill questions (see Appendix IId: Table 3).

Performance

We tracked the accuracy rates for questions for the required questions separately from the optional questions. (see Appendix IId: Table 4) Given that there is a larger number of responses for the required section, we ran Fisher's exact test and Chi-squared test against a null hypothesis that there would be no difference in the accuracies between the different question types. The results of both tests will be discussed in the conclusion.

Time to Answer

We also measured how long users took on average to answer each of the question types. However, due to the method of implementation within Qualtrics, we were unable to measure this for each respondent. Instead, the platform only provided us with summary metrics which may have been affected by outliers, particularly users who took a significant amount of time to complete the experiment. For example, 2 of the 36 users that completed the experiment took over 1 hour to complete the experiment, affecting the mean numbers below. Therefore, this data is provided merely for reference and will not be used to make inferences about the difficulty of the questions. (see Appendix IId: Table 5)

Experiment Findings

There were three high level hypotheses that we looked to validate through our experiment. Our first hypothesis, H1, was that users would perform much better on MC questions than on true/false or gap-fill questions. We made this assumption based on informal testing and usability tests that we had conducted previously. In order to test this hypothesis, we first tested to see if there was any statistical difference in performance between the different question types.

Both the Fisher's exact test and Chi-square test showed that we had a statistically significant difference between performance in question types. Therefore, we validated our belief that there is a difference in user performance between different question types. We sought to answer the question of whether multiple choice questions were statistically significantly easier to answer than the other question types. Appendix IId: Table 6 shows the data used to run the Chi-squared test and Appendix IId: Table 7 shows the standardized residuals that show that the probability of answering incorrectly on multiple choice questions was three standard deviations below the expected. Therefore, we also validated our hypothesis that multiple choice questions would be easier to answer than the other question types.

Surprisingly, we also noted that users were able to answer a higher percentage of gap-fill questions correctly than true/false questions. This may have been the effect of an availability heuristic in asking users to choose amongst a set of options versus recalling a specific detail from the article from memory. Users may have searched for the article using Google in order to answer the gap-fill question whereas they may have not done the same with true/false questions.

Secondly, we hypothesized, H2, that users would prefer multiple choice questions over gap-fill and true/false. As described in the previous section, 66% of our sample of users preferred multiple choice questions while 25% and 8% of the users preferred true/false and gap-fill respectively.

Finally, our last hypothesis, H3, was that there would be an inverse relationship between accuracy and the time taken to answer a question. Given that we were not asking questions in a setting where the score was important, we hypothesized that people spending more time on a question didn't necessarily mean that they were more likely to get it right. Furthermore, spending more time on a question may indicate that a question is more difficult and, therefore, may have lower performance.

We recorded the accuracy and time taken for each question, however, due to the format of the export data from Qualtrics, we were unable to perform the necessary statistical test for

correlation. However, in comparing the mean time taken to answer the question, users were marginally quicker to answer multiple choice (12.8s) than true/false (13.5s) but took slightly longer with gap-fill (18s). This difference could also be owed to the fact that they needed to type an answer only for a gap-fill question. Given the sample size and possible effect of outliers, we decided to not draw any inferences or make any claims from this statistic.

Limitations

Our users were taken from a convenience sample and may not have been sufficiently randomized. We were also not able to control for certain external biases such as the possibility of a user having read our sample article previously. The topic of article and type of question could have added some bias despite our efforts to normalize by choosing articles from different, but general, domains.

NLP Research

We used techniques and tools from natural language processing (NLP) to create our questions and explored the state of the art. Previous work in automatic question generation have extensively focused on the domain of education (Kunichika et al., 2004; Boyer et al., 2010). The majority of the work undertaken in designing, testing, and evaluating the Gadfly Project was founded on the work of two papers, *RevUP:* A*utomatic Gap-Fill Question Generation from Educational Texts* by Kumar et al. and *Question Generation via Over-generating Transformations and Ranking* by Heilman and Smith (2015; 2009 and 2010).

RevUP: Automatic Gap-Fill Question Generation from Educational Texts

This paper presents a new technique for generating gap-fill questions where the author uses a topic distribution model for shortlisting the sentences that serve as the basis for the question. The author then identifies gap-phrases from these selected sentences and uses Amazon Mechanical Turk (AMT) to evaluate the relevance of the various phrases. These labels are then combined with the sentences to create a dataset which is used to train a discriminative classifier. This paper talks about the use of questions in an educational setting and uses Campbell Biology, a biology text, as the corpus.

To reduce bias, the authors hand-picked sentences to ensure a mix of topics, sentence-lengths and gap-phrase lengths were present. AMT was then used to crowdsource evaluate a subset of 200 of these sentences and rankings were collected for 1306 gaps in total. Each of the raters were asked to judge whether the distractors, incorrect answers in a multiple choice (MC) question, for each question were good or bad on a scale of 1-9 and thus all of the sentences

were evaluated by three separate raters. Distractors with a score greater than 6 were considered good and this data was used to train a SVM model.

In order to evaluate the overall distractor selection process, the authors ran another test on AMT. This time, they evaluated 75 sentences with 300 distractors and asked raters to assign a score of 1-5 for each distractor. All scores greater than two were considered "fair" and their method led to 94% of the distractors being considered fair.

Question generation via Over-generating Transformations and Ranking

Heilman and Smith describe a process of overgeneration and ranking to create a set of questions of an acceptable quality. While their end product is wh-questions (such as who, what, when, where, why), an intermediate step in this process is to generate gap-fill questions. They took the help of 50 native English-speaking university students to rate questions as 'acceptable' or 'unacceptable' based on a set of predefined criteria. This data was then used to train a discriminative question ranker. The judgement criteria used both semantic and syntactic features that looked at the grammatical correctness of the questions as well as the quality and relevance of the answers.

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

Table showing possible deficiencies that were tracked by Heilman and Smith.

The authors evaluated their method by measuring the acceptance rate in the top 'x' percentage of their ranked questions and the total percentage of questions that were found to be 'acceptable'. Their setup had two possible conditions for presenting a set of questions to a user - randomly ranked questions and a set of questions which had been labelled as 'acceptable' by their classifier. The test also took into consideration that how the questions are ranked could vary based on the classification model so had tests run on three versions of the model, which were different based on the features, and compared the results between them. On their best model, the results found 27.3% of all test set questions were acceptable and 52.3% of the top 20% of ranked questions were acceptable.

They also ran an ablation experiment where they study the effects of removing each of the various types of features. They measured the percentage of acceptable questions within the top 20% and top 40% for different permutations of the feature types.

These two papers informed our approach and highlighted challenges to we would have to confront.

Design

While conducting our various lines of research we began designing both use cases and the question generation system - as the designs would both feed into and be fed from our research. The use cases included conversational user interface (UI) design and the system included both NLP components and the design of the application program interface (API). The later would provide the service in the Gadfly Project.

Gadfly Project - The Use Cases

In imagining and designing use cases we started with lessons from our exploratory interviews and iterated on the feedback from our user research and assessments of the technical capabilities of our system.

During the course of our research and design one team member followed the "working backwards" approach he learned while interning last summer at Amazon.com, Inc. by writing what he imagined The Gadfly Project could be. This helped our team communicate in more concrete terms. Publishing his vision[6] also served to elicit more comments from our community. Here is his one sentence tagline:

> A simple service that instantly enables developers to automatically generate questions from input text.

Comment-CAPTCHA

Our service needed an identified use case to demonstrate the utility of the service and to serve as a foil for its design. The initial use case was a comment-CAPTCHA service. A CAPTCHA is a type of question used on many web sites to confirm the user is a human by asking for a task to be completed that current machine intelligence finds hard. They are used to serve resources only to human visitors of the website. The comment-CAPTCHA service that we envisioned would, analogously, ask a question of persons intending to comment on a news article to distinguish those who read the article and developed a basic understanding of it from those who did not.

We imagined this originally as an effort to improve discourse around news on several levels. We believed the questions may provide a sort of pre-moderation of comments, improving

[6] Available on the internet at this link https://medium.com/@n1khl/the-gadfly-project-19a901bd5abe and at the end of this report: Appendix I: The Gadfly Project on Medium by Nikhil Mane

comment sections, by providing a barrier. We also imagined that being prompted to answer a question may introduce closer engagement with the article and also improve recall of at least the item being questioned. The mockup below imagines an implementation of comment-CAPTCHA.

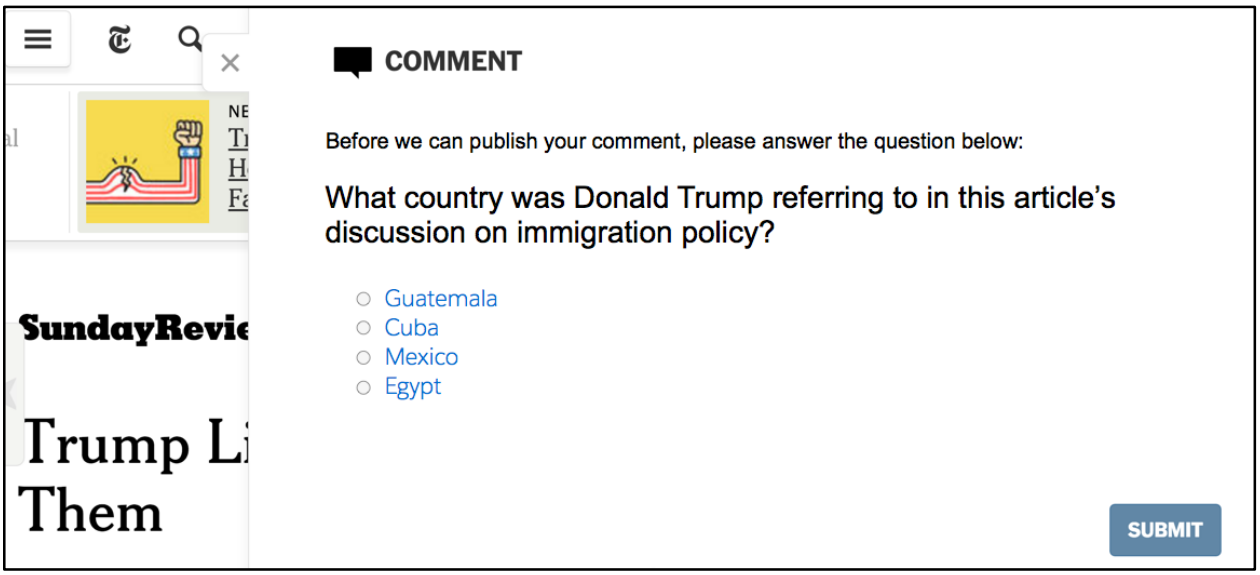


Figure 1. A mockup showing a user being faced with a multiple choice question before commenting.

As discussed above, this barrier may be seen as prohibitive to publishers and a source of friction for users. We also imagined that some implementation of such a service may have some performative purpose, condescending though it may be, in forcing readers looking to comment to consider whether they did or did not read the article. Our current thoughts about comment-CAPTCHA and imagined ways forward will be discussed below (see Future Work).

Twitter bots

We had considered creating a Twitter bot in order to display another use case for our automatic question generation service. This would serve three purposes. One, it would be a public display of our work in action. Two, it would be a test of the extensibility of our API. Three, it may draw attention from the news publishers who we are trying to work with.

Our planned approach was to create a Twitter bot that would automatically retweet every article that came from a source like the New York Times, and attach an automatically generated question from our service. This could be extended into several Twitter bots, one for each publication that we were interested in working with. In this way, it would be a public demonstration of the questions that are generated for the comment-CAPTCHA use case.

Our primary goal has been to increase the quality of questions that our system is generating. With this in mind, we could not think of a quick way to build a feedback mechanism in Twitter, given that our bots would require a followers before providing any use. So, in respect to time constraints and the overall goals of the project, we decided that this idea could be de-prioritized.

Slack bot users: QBot and TriviaBot

We were at a point in our project where we wanted to test our API to improve it. For this, we needed people to use the API and provide feedback on the questions being generated by the API. Slack provided a distribution mechanism suitable for our use cases. It provided easy access to several users (through the I School Slack) and allowed us to quickly gather feedback and iterate. The Slack bot users are essentially interfaces that allow users to converse with our API in a controlled manner.

Here, we have an initial wireframe of the conversational flow for QBot. A pain point we heard from our fellow classmates was how they couldn't keep up with all the news articles being posted in Slack, so we envisioned QBot as a way to collect and re-share articles at a later time. The idea was to have a bot listening for articles posted into various channels and then at set times ask questions generated from those articles. The general flow is shown here:

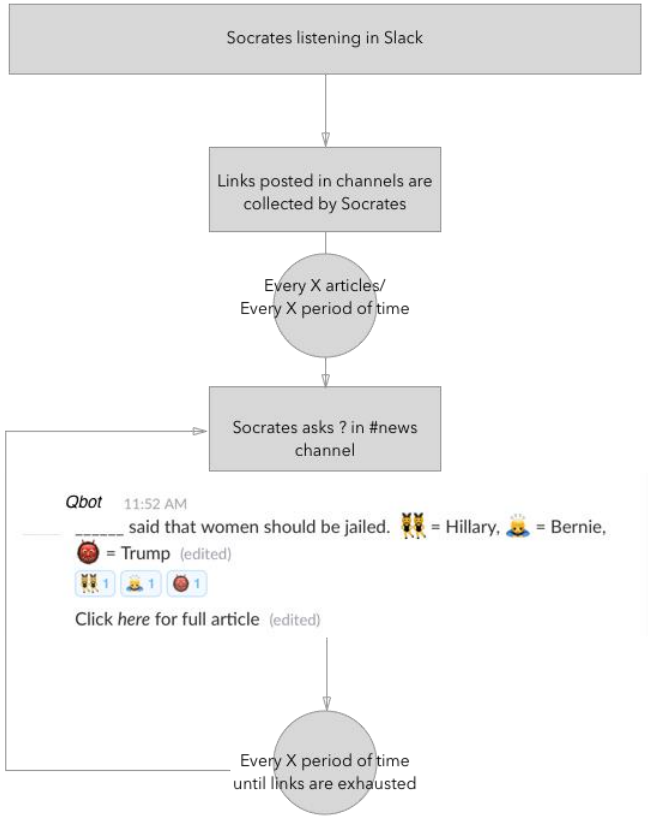


Figure 2. Use Case 1 Workflow

QBot The main motivation behind creating this was to learn about question quality and interaction flow. Users can interact with this bot to be quizzed on their topics of interest. Users send a URL to a news article as a message to the bot user. This triggers a conversation flow that leads to the user being quizzed on the content of the article. All the questions are generated using the gap-fill question generation capability of The Gadfly Project. The bot user occasionally asks for feedback on the quality of question generated. An example interaction is shown below.

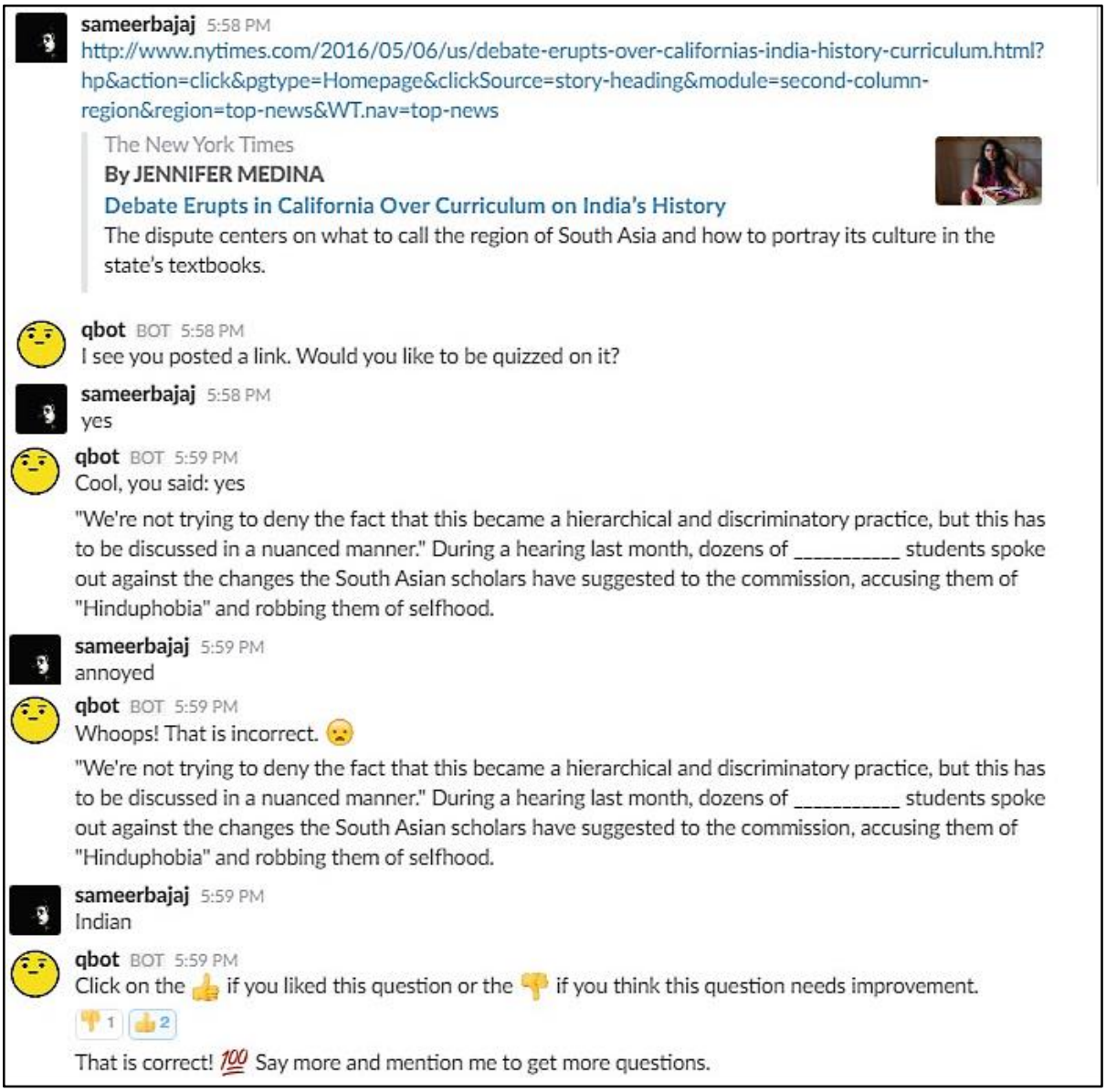


Figure 3. QBot being tested by a real user on 5/5/2016

Through our initial usability testing, we realized the need for a more natural use case. After discussion with our faculty advisor, we understood the potential for getting more feedback. This brings us to our next Slack bot user.

We imagined this as a different way to engage with the Slack community. The gamification of question solving could in theory draw people to look forward to the set time. This would ensure a continuous feedback cycle into our question generation system.

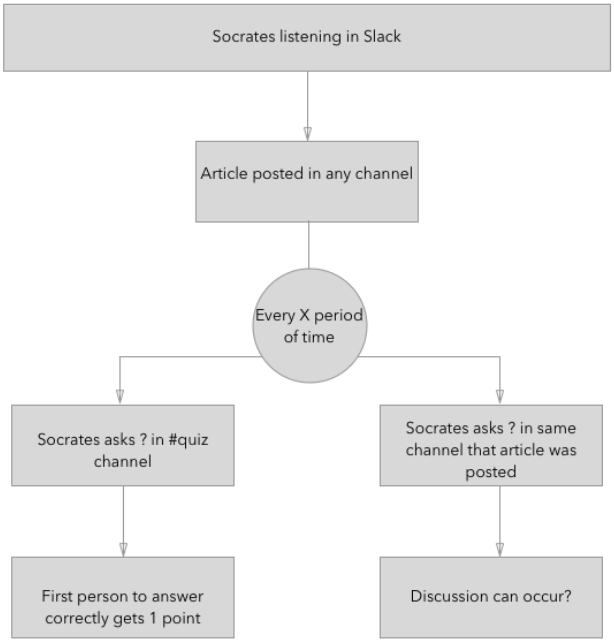


Figure 4. Use Case 2 Workflow

TriviaBot This bot user simulates the experience of playing news based trivia drawing upon the multiple choice question generation capability of The Gadfly Project. At a specific time (which you can set reminders for), TriviaBot messages the specific Slack channel with questions obtained from the Gadfly API. The messages include question text along with 4 answer choices. In the interest of reducing friction, the bot user adds emoticons corresponding to the choices as reactions to the message. The players can respond to questions by clicking on these reactions. The bot maintains state by counting responses, tracking time of response and maintaining a leaderboard. At the end of a trivia session, the bot messages the channel with these stats.

Figure 5. TriviaBot starting trivia after an automated prompt

We carried out in-person usability testing to get qualitative feedback and incorporated a feedback mechanism using the Slack API to get quantitative feedback.

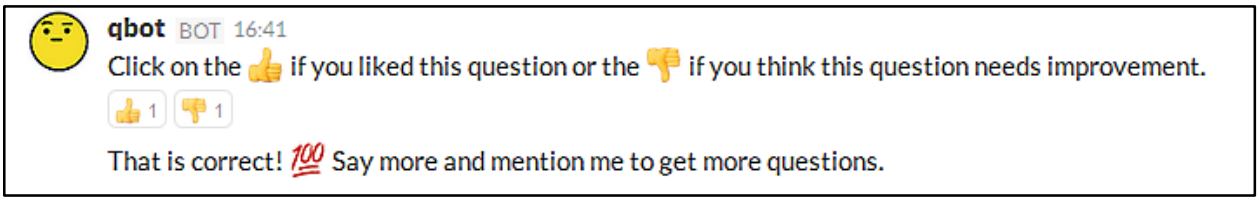


Figure 6. Gathering quantitative feedback using Slack's reaction feature.

Challenges

The problem space, as we envisioned it, is simply too large. Our preliminary research and initial user interviews confirmed our worries about the scale of this endeavor. The abuse that occurs in comment sections online and the legal, social, and technical complexities involved in addressing it are well-documented. Several online publications have already dedicated online real estate to announce why they are closing comment sections. In our generative user research phase, we faced hesitation and reluctance from our interviewees. They raised legitimate concerns about user experience. At the same time, we saw evidence for the utility of our goal — creating an extensible, automatic question generation service.

We wanted to be able to highlight the NLP work while providing a valuable and usable product in general. We also wanted to build in mechanisms so that our users could give us direct feedback into the quality of the questions being generated. Having these goals in mind allowed us to narrow down our use cases and maintain focus along the way.

Prototyping our designs also proved challenging as we chose not to Robin Hood[7] during any of our usability tests. We believe that foundation of the usability of our designs was the quality of the questions being generated, so all of our designs incorporated the real-time application of our engineering work. This raised technical glitches along with the regular design issues, but also allowed for quick iterations on technical development.

Conversational user interfaces (such as Slack bots) afford new ways of testing user interaction. They also bring new design challenges. With conversations, the possibilities of something going wrong are limitless. Our team did a lot of testing before we tested with other users to define certain outer bounds. Context is really important because the ways a conversation can go out of scope are myriad. One of our users tried several different ways of saying "yes" which is something we had designed for. However, another user replied with "si" and the bot could not

[7] Robin Hooding is a term used to describe a technique where a poor prototype imitates a rich implementation in usability studies through the targeted and secret intervention in the process by the experimenter. See also: Wizard of Oz technique.

respond to that because we hadn't considered language. For actual usability testing, it is important to set expectations with participants.

The primary goal of the usability assessment was to evaluate the current implementation of the API. However, we were also testing the platform it is built on (Slack) and the conversational user interface (the bot). It was important for us to remember that the "bot" itself is only one aspect of the user experience that we were testing. Having a clearly defined need and motivation at the start, helped us focus on what we actually needed to test -- the quality of the questions being generated. In addition to this, we also ended up observing the quality of interactions people were having with the interface.

Gadfly Project - NLP

NLP Platform

The core of the Gadfly Project is a natural language processing (NLP) system that takes raw text, performs a set of algorithmic functions, and returns a list of questions of the specified type (currently, multiple choice (MC) or gap-fill (GF)). The NLP tasks are reliant on spaCy, a general purpose NLP library which is an alternative to the more popular NLTK Python library. We rely on spaCy for certain tasks such as sentence segmentation, named entity recognition, and key sentence identification.

While our experience with NLP was with Natural Language Toolkit (NLTK), we ran into frustration with entity recognition and speed (Looper and Bird, 2006). At the suggestion of a fellow student we looked at spaCy.[8] Matthew Honnibal's spaCy is self-described as "Industrial-strength Natural Language Processing." Honnibal has written on the two platforms, including why he decided to write spaCy rather than contribute to NLTK[9], arguing that "NLTK was created to support education. Most of what's there is for demo purposes, to help students explore ideas." Whereas, "spaCy is written to help you get things done."[10]

The platform provides several features we made significant use of: sentence segmentation, named entity recognition, word frequencies in a large corpus (60 billion tokens (words)[11]), word embedding representations ("a dense real-valued vector that supports similarity queries

[8] Andrea Gagliano, in discussion, February 22, 2016.

[9] https://spacy.io/blog/dead-code-should-be-buried

[10] https://www.quora.com/What-are-the-advantages-of-Spacy-vs-NLTK

[11] https://github.com/spacy-io/spaCy/issues/10#issuecomment-124896129

between words”[12]), as well as tree parses that we experimented with for currently unimplemented features of the Gadfly Project. The platform is actively developed with a growing user base.

Approach

Gap-Fill Questions

The most fundamental question type in our system is gap-fill; both multiple choice and true/false questions are derived from gap-fill questions. Gap-fill questions are generated by segmenting input text into sentences using spaCy’s built-in sentence segmenter. We identify segments by assigning a score to each sentence. [VI] This score, discussed below in Interestingness, is generated by tokenizing words for each sentence, and taking the sum of spaCy’s unigram log-probability of each word (which is estimated from counts from a built-in corpus and smoothed using Simple Good Turing estimation). We take ten sentences with the lowest scores (lower being more important), from this process and, for each, replace named-entities (using spaCy’s named entity recognizer) of the following types: PERSON, NORP, FAC, ORG, GPE, LOC, PRODUCT, EVENT, WORK_OF_ART, LAW[13]) with a blank (“_____”). The named entity that is removed is the answer phrase, and the modified sentence is the resulting gap-fill question.

In some sentences, it is possible that there may be more than one named entity that we could potentially remove to generate the question phrase. In these cases, we originally created every possible question phrase. However, due to the confusion that this caused for users during our user testing, we decided to ensure that only one possible question phrase was generated from each sentence. In lieu of randomly selecting a named entity for each source sentence, we decided to use the New York Times Article search API to select the most “popular” named entity for each source sentence. We query the API for all the named entities, and search for the number of New York Times articles that mention that entity over the past 30 days and select the entity with the highest number of hits. This helps us overcome the possibility of selecting obscure entities from a given article, and instead helps us prioritize more popular entities, which may be desirable given the temporal effects evident in news.

The process to create gap-fill questions serves as the base for multiple choice question generation.

---

[12] https://spacy.io/docs

[13] NORP = Nationalities or religious or political groups; FAC = Buildings, airports, highways, bridges, etc.; GPE = Countries, cities, states; LOC = Non-GPE locations, mountain ranges, bodies of water.

Multiple Choice Questions

Multiple choice questions are created by using the answer phrase from the gap-fill questions, to generate a set of distractors, other answer choices, for each question. Distractors are generated by finding other entities of the same type (person, organization, etc.) from the article, as identified by spaCy, with some heuristic checks to improve the probability that the answer choices are distinct and realistic. The principal heuristic check is to run spaCy's similarity function with every other entity of the same type. This function finds the most similar token through a vector of values created from the corpus and acts as a word embedding representation. The quality of results are heavily dependent on the original corpus and do not truly indicate similarity in meaning so much as an indication that the tokens appear in similar contexts. For example, checking for the most similar tokens to 'vandals' returns the tokens 'spammers' and 'trolls'.

In some instances we have more refined heuristics. For location entities, primarily states in the United States and country names, we fall back to generating distractors from a static list. If a suggested distractor, the tokens with similar entity types, was not in the list it was removed. If there were not at least three distractors remaining, distractors were from the appropriate static list. For example, imagine an article mentioning the following geopolitical entities (GPE): Virginia, Alabama, Nevada, Detroit, Asia, and Liberia. If the correct answer phrase is Virginia, this heuristic would remove from consideration those GPE entities not listed in the US states static list and replace with one of the ten most similar US states. The static lists are a result of pre-processing where each such GPE token is compared to each other in the appropriate sub-category to and compared with spaCy's similarity function.

Overreliance on the entity type labeling within spaCy makes some selected distractors amusing or clearly incorrect. One example is how Twitter is alternately labeled, depending on local context, as a PERSON, PRODUCT, ORG, or GPE. This may seem wholly incorrect on the part of spaCy but in various contexts you can imagine how Twitter is referenced in those various ways and you can imagine why it would not be beneficial to hard-code a change such that Twitter is always treated as any one of those. For example, here are four examples from the New York Times:

> "In the wake of negative press, my relationship with Twitter switched instantly from love to hate."
> "Donald J. Trump is perhaps the most prolific user of Twitter in political history."
> "Twitter stock price spikes briefly after false $31 billion takeover rumor was reported."
> ""They'll make it so your mom can go to Twitter," he said."

The contextual labeling matters when the variety of other entity labels are presented as options. While humor is not something we decry, this issue may reduce how discriminating the questions can function and some combination of answer choices may prove offensive. We have not yet identified a way to adequately solve this, though envision blending the variety of entity labels for a single token within an article or a corpus of articles may prove fruitful.

Challenges and Limitations

Interestingness

After developing the ability to generate gap-fill questions we quickly realized that many of the questions were not interesting and use cases were limited to those wanting to test nearly eidetic recall. We then considered two approaches to identify interesting sentences. Initially we considered and implemented an NLP summarizer and selected sentences from the summaries but that proved inadequate. We also considered implementing a tf–idf based approach but quickly realized the amount of signal in an individual sentence was not enough to distinguish interesting sentences within an article (the use in finding interesting documents in a collection does not scale down well) especially when particular terms were not generalizable as interesting for most news articles.

We then implemented a metric that identified segments by selecting the segment with the lowest sum for its five least probable tokens. Creating a metric from only the lowest five served to normalize for longer sentences. While in some sense providing interesting sentences, this selected for particularly distinct segments with exceedingly rare or out of vocabulary words. The usability studies indicated users did not think some of our questions were particularly important.

Our current segment identification implementation tries to get at both interesting and important. We now retain the lowest sum for the five least probable tokens but identify those tokens after first running the segments through several filters. The token is not considered if the spaCy determined entity type for the token is PERSON, if the token appears to be an email or a Twitter handle, or if the token is not in the spaCy corpus. If the token entity type is MONEY, CARDINAL, or QUANTITY, or if the log-probability meets an identified rarity threshold, the log-probability of that token is adjusted higher. Lastly, the score is incremented higher by the segments index within the article. That is, the later in the article a segment appears, it is marginally less likely to be selected.

Without a singular use case it is difficult to optimize on or even identify a particular operationalization of a relevant segment. When designing for a comment-CAPTCHA system, we identified an experiment that could be performed to optimize segment selection for a particular purpose. We did not conduct the experiment but anticipated having two groups of readers exposed to the same questions after one of two conditions, either reading an article or only reading the headline of the article. We would then train a model to classify those segments that produced questions with the best fit for correct answers in the read-article group while the incorrect answers in the read-headline group. It is not clear that such a test would help optimize the Gadfly Project on the [QBot] or [TriviaBot] use cases not what optimization strategy might be pursued for either.

Fuzzy matching

In order to have end-to-end support for gap-fill questions we had to consider how we could support the validation of written answers. One of the challenges with evaluating gap-fill questions is in defining the threshold for accepting answers. It was important to allow for a certain level of error from the user input but not enough to accept a completely wrong answer. There are several metrics[14] that have been traditionally used for this purpose of measuring string similarity. We decided to use an in-built Python library *difflib* as it gave us sufficiently good performance.

Segmentation

There are some limitations in the default sentence segmentation from spaCy, particularly when quotes are in the sentences. Disjointed quotes were not useful for our purposes so we developed rules to identify all sentences within one quotation and join them together. We now identify these segments for our questions.

This rule based system is not perfect. An example of difficulty in determining some sort of useful segmentation is provided here and results from the lack of perfectly controlled conventions and typos. Here is one example, an elided quotation mark (marked in red), from a New York Times article that led to considerable consternation in improving the segmentation process but resulted, in the end, in more robust code:

> "During our meeting, I saw the brilliant, opinionated, focused, generous — and privacy-seeking — person that matches the Satoshi I worked with six years ago.**"**

[14] http://web.archive.org/web/20081224234350/http://www.dcs.shef.ac.uk/~sam/stringmetrics.html

> But the Bitcoin community was not in consensus.
>
> Another of the leading developers working on Bitcoin's basic software, Gregory Maxwell, said that the evidence presented by Mr. Wright was not enough to convince him.[15]

There are still many issues with sentence segmentation, but now it is largely the combining of two sentences, rather than splitting sentences at quotation`` marks. This is acceptable in our current implementation.

Heuristic-rules

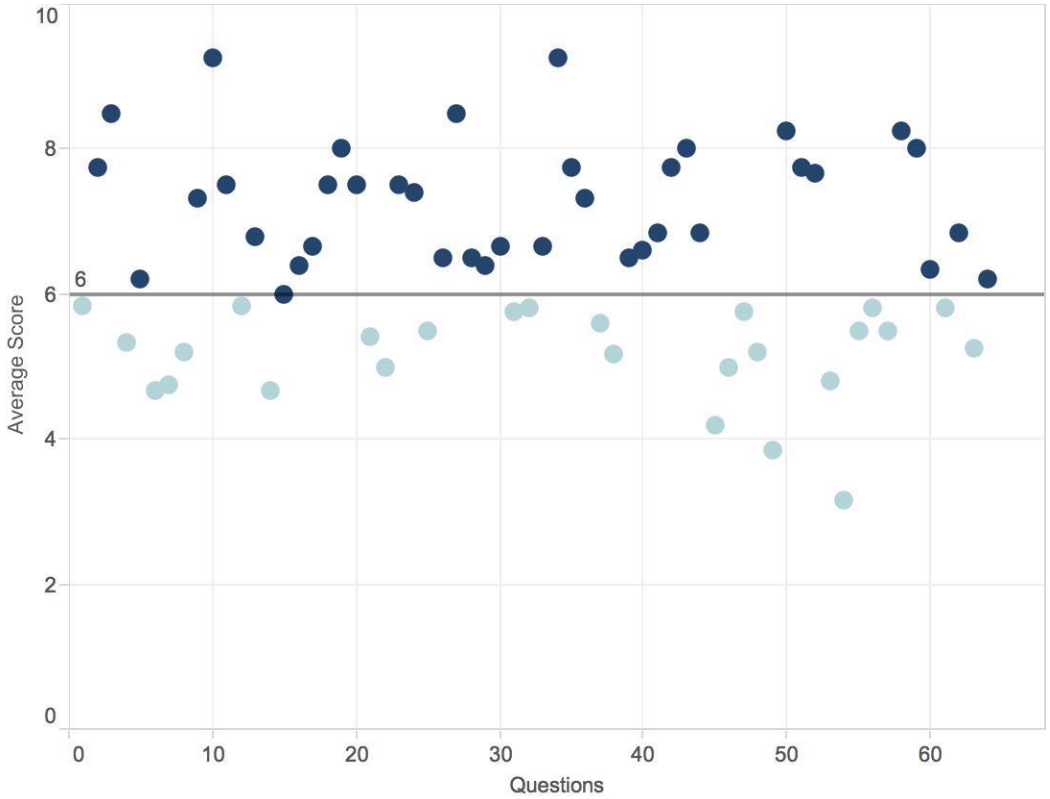
While we have extracted features, and stored them in sentence objects, that may be useful to evaluate our selected sentences and generated questions, we currently use them only in rule-based heuristics. If provided a large and varied population of users on a large and varied collection of articles a supervised learning approach may prove useful.

[15] http://www.nytimes.com/2016/05/03/business/dealbook/bitcoin-craig-wright-satoshi-nakamoto.html

Evaluation

In addition to the various evaluations reported on in the User Research section, we conducted an evaluation of our current state of question generation. We took eight recent New York Times articles and automatically produced eight multiple choice questions for each (for a total of 64 questions). We used the Qualtrics online survey platform to present evaluators the article headline, the New York Times one-line description of the article[16], and those eight questions. Nine evaluators, from a convenience sample, were each randomly presented four of the articles (the headline, description, and questions) to evaluate on a scale of 0 to 10. They were provided the following instructions:

> You will be asked to provide a subjective rating (from 0 to 10) on the quality of each question.
> This rating of quality should be based on your perception of:
> - importance or interestingness of the content tested in the question
> - effectiveness of the alternate answer choices



Visualization 1.

[16] See the content of the description meta tag in the source of any New York Times article.

The results of the evaluation are encouraging. The mean across all evaluators and questions was 6.45. 57.8% of the questions had a mean rating above 6 (acceptable); 9.4% above 8 (gold standard).

Future Work

The group made a significant effort to establish a strong foundation for a question generation service with the design of the technical architecture for the project (see Appendix III). We believe that we have laid the groundwork to quickly implement new forms of question generation with the core Gadfly Project NLP and have allowed the possibility of testing various use cases of a question generation service through our web API. To this end, individual group members have expressed interest in continuing development in one or both of the services.

The web interface, GadflyProject.com, has been developed to expose the inner workings of our NLP process. We would like to make it interactive, allowing readers to not only see identified segments but click to see the questions that could be created from segments not selected by our system. This could also be a platform for incorporating feedback into a learning system to better select both segments and questions.

The team will continue testing various use cases of automatic question generation including different uses of Slack bots. This allows for testing interactions either with one user at a time, or multiple parties at once, as well as testing in different contexts. We would also like to go back to the newspaper publishers to report on our findings to see if that would convince them to test our comment-system. The team has already reached out to larger publishers and are hopeful of starting conversations with them. We would also love to continue interviewing habitual commenters to learn more about how they interact with digital publications. This would allow us to build the smoothest flow of leaving a comment in order to reduce as much friction as possible with the introduction of our question generation system.

Conclusion

In addition to our working API for automatic question generation, this project has made several contributions. Our interviews revealed that the idea of automatically-generated questions inspires creative use case imaginings. While publishers were quick to identify risks of added friction and of appearing condescending, they also quickly understood what we were exploring and thought it an interesting problem. Our survey indicated that asking readers questions on the content of a news article may be a more viable component of a larger comment moderation system than publishers assumed. Our usability studies showed that automatically-generated questions on news articles can be engaging. Our question evaluation showed that our automatically-generated questions are interesting and important.

Our exploration in this project has shown us that several imagined use cases for the automatic generation of questions from news articles are new and ripe for further exploration. Our web API allows users to quickly test various use cases of automatic question generation without forcing them to implement it from scratch.

While we acknowledge that we have only scratched the surface of automatic question generation with our own technical implementation, we believe that we have showcased the potential of a service such as this. This, despite the fact that our own foray into this domain is as relative newcomers. With additional contributions and input on the question generation process as well as its potential use cases from the community, we are more likely to see the realization of its full potential.

Works Cited

Bird, Steven. "NLTK: the natural language toolkit." Proceedings of the COLING/ACL on Interactive presentation sessions. Association for Computational Linguistics, 2006.

Boyer, Kristy Elizabeth and Piwek, Paul eds. (2010). Proceedings of QG2010- The Third Workshop on Question Generation. Pittsburgh- questiongeneration.org

Citron, Danielle Keats. Hate crimes in Cyberspace. Harvard University Press, 2014.

Heilman, Michael, and Noah A. Smith. *Question generation via overgenerating transformations and ranking*. No. CMU-LTI-09-013. CARNEGIE-MELLON UNIV PITTSBURGH PA LANGUAGE TECHNOLOGIES INST, 2009.

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

Iyengar, Shanto, and Sean J. Westwood. "Fear and loathing across party lines: New evidence on group polarization." American Journal of Political Science 59.3 (2015): 690-707.

Jeong, Sarah. The Internet Of Garbage. Forbes Media, 2015.

Kumar, Girish, Rafael E. Banchs, and Luis F. D'Haro. "RevUP: Automatic Gap-Fill Question Generation from Educational Texts." Silver Sponsor (2015): 154-161.

Kunichika, Hidenobu, et al. "Automated question generation methods for intelligent English learning systems and its evaluation." Proc. of ICCE. 2004.

Lat, David. "Comments Are Making the Internet Worse. So We Got Rid of Them." Washington Post. The Washington Post, 21 Apr. 2016. Web. 05 May 2016.

Loper, Edward, and Steven Bird. "NLTK: The natural language toolkit." *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*. Association for Computational Linguistics, 2002.

Pew Research Center, "Political Polarization and Media Habits." Oct. 2014.

Pew Research Center, “A Wider Ideological Gap Between More and Less Educated Adults.” Apr. 2016.

Santana, Arthur D. "Virtuous or Vitriolic: The effect of anonymity on civility in online newspaper reader comment boards." Journalism Practice 8.1 (2014): 18-33.

Appendices

Appendix I: The Gadfly Project on Medium by Nikhil Mane

A simple service that instantly enables developers to automatically generate questions from input text.

Today, the VANDALS announced the release of the Gadfly[17] Web API that powers the Twitter account, @question_exe, which has been tweeting intriguing questions about articles in the New York Times. The API unifies a set of tools that make it easier for developers to generate questions based on input text without the headache of dealing with NLP libraries. The diligent Vandals team plans to provide support to make integration as seamless as possible.

The problem of question generation from text is an interesting one addressed by Heilman and Smith in 2011. The VANDALS team, building on that work, has created a web service that uses proprietary algorithms to process text from a news article to generate questions about that article.

“Richard Feynman famously said, ‘There is no learning without having to pose a question.’ This is our attempt at posing questions based on the news to inform readers,” said Daniel Griffin, one of the creators of The Gadfly Project.

The team thinks there are several possible applications for such a service. One of which is their conversational interface, QBot. “It is like nothing I have ever used. It’s made reading news much more interesting,” said Steve, one of their beta testers. Simply put, QBot asks you questions about the news. “Sometimes, it’s just like fill in the blanks, you know. I personally like to use it as a trivia app. Time just flies on BART,” said Jenn, another beta tester. The team is looking to launch this in a few months.

For now, you can visit http://GadflyProject.com. Paste a news article URL in the text box and start asking questions.

“We were interested in the idea of using question generation as a way to improve online discourse around news. Our initial idea was that we would generate questions based on the

[17] A gadfly is a person who interferes the status quo of a society or community by posing novel, potently upsetting questions, usually directed at authorities as per Wikipedia. The term is originally associated with the ancient Greek philosopher Socrates, in his defense when on trial for his life.

news article to prevent people who haven't read it from commenting on it. However, we found out through user research and customer interviews that this problem is larger than we envisioned. The news publishers that we spoke to were interested in the idea but they had reservations about adopting it. We conducted a public brainstorming session during InfoCamp 2016 and realized that question generation itself is quite an interesting problem. We decided to make the process [using algorithms to generate questions from text] easy and scalable. We have had interest from some companies in the education and news sector. We are genuinely excited about exploring the potential uses for our service," the team said in a joint statement.

—

We were at an odd stage[18] in our capstone project (we had done work towards achieving our goal but our success criteria was not well defined). As a team, we decided to work out what it is that we wanted to create as deliverables by the end of our semester. I decided to use the "working backwards" approach which I learnt over my summer internship at Amazon. You work backwards from the customer to identify what you want to build. A product manager, ideally writes out a document that lists out the customer problem, how current solutions fail and how the new one will succeed. I modified this format to talk about what we were creating (since there are no current solutions that do this), potential benefits we would create for our users along with use cases that we could actually present as deliverables. We are currently hard at work improving our API and our Slack bot users. Stay tuned for more updates on that!

Appendix II: User Research

Appendix IIa: Interviews

Interview Protocol

- Who we are and generally what we are working on.
- Tell us about the comment process
    - Technically what happens
    - How are the comments moderated, if they are at all
- How important is commenting to your publication?
    - Does it generate revenue for you?
    - Would you want to get better data about the content of the comments?
    - Would you rather have more comments or higher quality of comments?
- Give more details about our automated question generation
    - What issues do you see?

[18] This was published online on April 12, 2016 and written a week before.

- Is this something you would try to implement?
- Possible issues that we are aware of
  - Changing type of comment left
  - Reducing comments
  - Causing bias
  - Quality of questions

Interviewees

Lance Knobel, Co-Founder, Berkeleyside:

Alison Fu, Online Managing Editor, Daily Cal:

Katie Dowd, Senior Producer, SF Gate:

Daniel Ha, CEO, Disqus

Course Hero, Nancy Chan, PM and Dayne Bratsman, Technical Lead for Q&A Team

Tamara Straus, Freelance Journalist (Quartz, Mother Jones, San Francisco Chronicle)

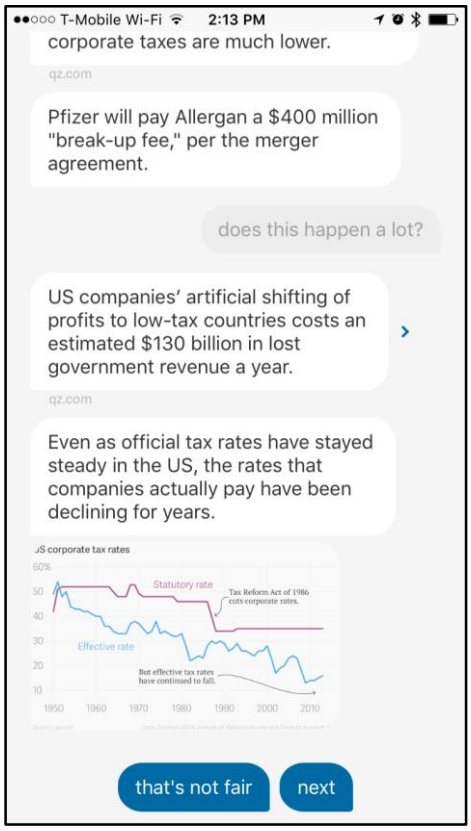
Appendix IIb: Survey

The survey consisted of the following questions:

- How many news articles do you read each week?
- How often do you comment on a news article each week?
- How often do you read the comment section of a news article each week?
- How do you usually feel when reading over a comment section for an article?
- If you are frustrated, angry, or generally upset when you read over a comment section, what usually causes this feeling?
- Is there a newspaper publisher whose comment section you respect more than others? If so, please list:
- If you read comments in the NYTimes, which section of the comments do you read more of?
- Is there a publication's comment section that you particularly enjoy reading? If so, please identify:
- [Given the comment-CAPTCHA example, see Figure 1 above.] Would you proceed to answer the question in order to make your comment?
- If no, why wouldn't you want to answer the question?
- Would you login with some form of credentials (Facebook, Google, Twitter, Email) in order to make a comment?

- If no, why wouldn't you use your social media credentials to login?
- Would you rather answer a question based off of the article, or login using social media credentials in order to leave a comment?
- Why would you choose one method over another?
- Have you made a comment on an article in the last month?
- Did you have to log in to make a comment?
- What was the publication that you left a comment with?
- For the article that you left a comment on, did you also share it via social media?
- Do you read more of your news from your social media feed, directly from a publisher, or through an email digest/newsletter?
- If you get your news from social media, what would you do if you saw a "Verified Reader" badge like this:

- Why would you lean towards reading it?
- Would you enjoy receiving news through a messaging app? Here is an example from Quartz:



- Do you use Slack?
- Would you interact with a Slack bot that asked questions based off of the news stories posted to a channel?
- If the Slack bot asked you a question related to current events, would you…
    - Find it annoying
    - Try to answer the question
    - Ignore it
    - Other
- Imagine that you could earn points when you answer questions related to current news. Would you choose to answer the questions?
- Imagine receiving articles via a weekly newsletter. Instead of reading a short blurb for each article, the newsletter is compiled with questions relating to the news of the week. Would you choose to answer the questions?
- If you didn't know the answer to the questions, would you click on the link which would bring you to the article that contains the answers?
- We are currently developing a custom slackbot that can interact with the news. If you are curious, or would like to help us test it out, please leave your email address here:

Appendix IIc: Usability Tests

General:

- How to ask them a question: pick an article from the NYT that you think you know and you'll be quizzed on it.
- Have tab open with the database of questions/answers so we can step in if they don't know the answer
- Run test in different ways of whether we add ABCD, reacting, or typing full answer

Setup:

- Make sure we are testing people who use Slack.
- Video recording
- Note taking
- If video isn't acceptable, ask for audio recording.
- Invite the tester into channel, either private channel in I School Slack, or gadfly-project Slack

Script:

- Welcome the user and give a brief explanation of how this is work for a final project. State that we created a bot that can generate questions, but don't give much detail.
- Currently, the bot is a proof of concept that we can generate questions automatically.
- Say it's still a work in progress, so thinking aloud as they are interacting with the bot would really help. Make sure they understand it is NOT their fault if something goes wrong.

Tasks:

- Go to NYTimes, pick an article that you think you know the subject matter.
- Go to Slack.
    - Post the url in channel: _____. -OR- Direct message @qbot with the url
- Mark a question as a good or bad question.
- Pick another article that you haven't read and know very little about and paste into the Slack channel.

Appendix IId: Experiment

Articles and Generated Questions

The names of these articles and questions generated are provided here as reference to the claims made about them and as examples of gap-fill questions (including the various grammatical errors produced).

Bernie Sanders Campaign Hopes an Endorsement Resonates in New York - The New York Times

Q: As Gov. John Kasich's team has turned its attention to Senator Ted Cruz , the " super PAC " supporting Mr. Kasich, New Day for America, has been parroting some lines from Donald J. Trump , calling the Texas senator "_____" and derisively criticizing his "New York values" comment., A: Lyin' Ted

Q: Mr. Sanders will need that help, as he faces an uphill battle in the _____ primary: The most recent Quinnipiac University poll in the state had him trailing Ms. Clinton by 12 percentage points., A: New York

Q: The endorsement of Ms. Garner didn't help Mr. Sanders in _____ , where Mrs. Clinton beat him fairly soundly — by more than six to one — among black voters., Choices: {'Texas', 'South Carolina', 'Arkansas', 'North Dakota'}, A: South Carolina

Q: Ms. Garner then delivers the ad's most important line — "I believe _____ is a protester" — and an officer slams the door on a police van., Choices: {'Donald J. Trump', 'Eric Garner', 'Bernie Sanders', 'Rand Paul'}, A: Bernie Sanders

Q: Changing channels … Mr. Sanders 's campaign has swapped state-specific footage for its "America" ad— featuring the Simon and Garfunkel song — for most earlier contests., Choices: {True, False}, A: True

Q: And Mr. Cruz will need that help, as he faces an uphill battle in the New York primary: The most recent Quinnipiac University poll in the state had him trailing Ms. Clinton by 12 percentage points. Choices: {True, False}, A: False

Egypt Gives Saudi Arabia 2 Islands in a Show of Gratitude - The New York Times

Q: Saudi Arabia transferred Tiran and Sanafir to Egyptian control in 1950 amid concerns that _____ might seize them. A: Israel

Q: "It seems to many _____ that the president is selling land for Saudi riyals." Mr. Sisi has faced unusually sharp criticism recently for his handling of the struggling economy and the death of an Italian graduate student, Giulio Regeni., A. Egyptians

Q: A small demonstration erupted in _____ , the site of the 2011 protests that led to the ouster of President Hosni Mubarak., Choices: {'Syria', 'Saudi Arabia', 'Yemen', 'Tahrir Square'}, A: Tahrir Square

Q: In a flood of social media posts , critics called Mr. Sisi " _____ ," referring to a character in an old Egyptian song who had sold his land — a shameful act in the eyes of rural Egyptians. , Choices: {'Al Azhar', 'Nathan Brown', 'Hosni Mubarak', 'Awaad'}, A: Awaad

Q: Egypt's cabinet announced on Saturday that it was transferring sovereignty of Tiran and Sanafir, arid and uninhabited islands at the mouth of the Red Sea, to Saudi Arabia., Choices: {True, False}, A: False

Q: All of a sudden, everyone is acting as if they were vacationing there, when none had gone anywhere near it," the television presenter Hosni Mubarak said in an impatient outburst on his show. , Choices: {True, False}, A: False

Why Apple’s Stand Against the F.B.I. Hurts Its Own Customers - The New York Times

Q: Ultimately, the question is this: For lawful access to material important to terrorism investigations, would we rather trust _____ itself under the close supervision of the courts, or the F.B.I. and some private company that makes money selling cellphone hacks?, A: Apple

Q: The F.B.I. has already found a company able to access Mr. _____’s phone without Apple's assistance, presumably taking advantage of a vulnerability that Apple has either not yet identified or not yet patched., A: Farook

Q: While the _____ has refused to say whether it will ultimately share the vulnerability with Apple (and Senator Dianne Feinstein, Democrat of California and vice chairwoman of the

Senate Intelligence committee, has suggested that it should not), Apple's previous position is almost certain to make the government more likely to withhold the information., Choices: {'the White House', 'the Brookings Institution', 'the Justice Department', 'F.B.I.'}, A: F.B.I.

Q: First, as Ben Wittes of _____ has pointed out, the F.B.I. has no legal obligation to disclose the vulnerability., Choices: {'the Federal Bureau of Investigation', 'the White House', 'the Justice Department', 'the Brookings Institution'}, A: the Brookings Institution

Q: Apple is now asking the F.B.I. to "responsibly" disclose the vulnerability so that Apple can rapidly patch it. Choices: {True, False}, A: True

Q: On Friday, the F.B.I. again sought Apple 's assistance — this time to help crack an iPhone belonging to a convicted drug dealer — by requesting that a federal judge overturn an earlier decision in Brooklyn supporting Apple. Choices: {True, False}, A: True

The Islamic State of Molenbeek - The New York Times

Q: Abdelhamid Abaaoud, the suspected chief planner of the Madrid attacks, lived in Molenbeek. Choices: {True, False}, A: False

Q: The large-scale immigration from Turkey and North Africa that began a half-century ago at a time of economic boom has — at a time of economic stagnation — led to near-ghettos in or around many European cities where the jobless descendants of those migrants are sometimes radicalized by Wahhabi clerics. Choices: {True, False}, A: True

Q: After the carnage in Paris and Brussels, the laissez-faire approach that had allowed those clerics to proselytize, private Muslim schools to multiply in _____ , prisons to serve as incubators of jihadism, youths to drift to ISIS land in Syria and back, and districts like Molenbeek or Schaerbeek to drift into a void of negligence, has to cease., Choices: {'Belgium', 'Malta', 'France', 'Russia'}, A: France

Q: Salah Abdeslam, the only surviving direct participant in the _____ attacks , hid in Molenbeek before his arrest on March 18., Choices: {'New Mexico', 'Paris', 'France', 'Russia'}, A: Paris

Q: Yet even today there's something soporific about this French -speaking city marooned within Flemish-speaking Flanders, beset by administrative and linguistic divisions and the lethargy that stems from them, home to a poorly integrated immigrant population of mainly Moroccan and Turkish descent (41 percent of the population of _____ is Muslim), and housing the major institutions of a fraying European Union. A: Molenbeek

Q: _____ — a hodgepodge of three regions (Flanders, French-speaking Wallonia and Brussels), three linguistic communities (Flemish, French and German) and a weak federal government — is dysfunctional., A: Belgium

Tables

Combination	First Article	Second Article
A	1	3
B	1	4
C	2	3
D	2	4

Table 1: Possible combinations of the articles that were shown to the user

Combination	First Article	Second Article	# of Responses
A	1	3	12
B	1	4	7
C	2	3	7
D	2	4	10

Table 2: Number of responses for each combination of articles

Combination	# of Responses	# Preferred Multiple Choice	# Preferred True/False	# Preferred Gap-Fill
A	12	9	2	1
B	7	6	1	0
C	7	1	4	2
D	10	8	2	0
Total	**36**	**24**	**9**	**3**

Table 3: User preference for question types when making a decision after the required section

Combination	Section	MC Correct %	T/F Correct %	Gap Fill Correct %
A (1 & 3)	Required Question Types	96% (23/24)	83% (20/24)	83% (20/24)
	Optional Question Type	72% (13/18)	100% (4/4)	100% (2/2)
B (1 & 4)	Required Question Types	100% (14/14)	57% (8/14)	71% (10/14)
	Optional Question Type	66% (8/12)	0% (0/2)	NA
C (2 & 3)	Required Question Types	86% (12/14)	36% (5/14)	79% (11/14)
	Optional Question Type	100% (2/2)	100% (8/8)	100% (4/4)
D (2 & 4)	Required Question Types	95% (19/20)	45% (9/20)	70% (14/20)
	Optional Question Type	69% (11/16)	50% (2/4)	NA
Total	**Required Articles**	**94% (68/72)**	**58% (42/72)**	**76% (55/72)**
	Optional Article	**71% (34/48)**	**78% (14/18)**	**100% (6/6)**

Table 4: Raw performance data across the various question types and required and optional articles.

Question Type	Min	Max	Mean	Count
Multiple Choice	3.1 Seconds	105 Seconds	12.8 Seconds	126
True/False	3.7 Seconds	42 Seconds	13.5 Seconds	91
Gap-Fill	3.6 Seconds	98 Seconds	18.1 Seconds	79

Table 5: Response times across the various question types and required and optional articles.[19]

	MC	true/false	gap-fill

[19] Due to the export from the timing function on the Qualtrics platform, we are unable to calculate some metrics such as median and standard deviation.

Correct	**68**	**42**	**55**
Incorrect	**4**	**30**	**17**

Table 6: Entry data for Chi-Square test.

	MC	**true/false**	**gap-fill**
Correct	**+1.75**	**-1.75**	**0**
Incorrect	**-3.15**	**+3.15**	**0**

Table 7: Standardized residuals.

Appendix III: Implementation

We wanted the Gadfly Project to be extensible and support a potentially diverse set of use cases, many of which are described in detail in this paper. In order to do so, we decided from the onset to be very careful about the design of the technical architecture of the project. The core of the project is the set of algorithms that compose the NLP library, referred to as the Gadfly Project. However, we wanted to provide automatic question generation as a web service. Therefore, we built a lightweight API framework using Python's Flask library. This web API wrapper, referred to as the Gadfly Web API[20], uses the Gadfly Project (NLP) library but also provides some of the functionality relevant to enabling it as a web service. This includes the different endpoints to provide more insight into the intermediate steps of the question generation process such as the sentence segmentation, key sentence identification, etc. Furthermore, by separating the API and NLP tasks, we are able to provide the automatic question generation functionality as both a Python library and as a web api.

The NLP required to automatically generate question can be an opaque process and end-users do not have much insight into the steps necessary to generate questions. We believe that this goes against the core of how we want to design this system. In order to showcase how the NLP system operates and to assist internal testing, we built a simple web application, referred to as Gadfly Web[21], that allows users to use the web api through a gui interface. In designing our services in this manner, we are able to more thoroughly test our system because we are consumers of our own service[22].

[20] API documentation here https://github.com/TheGadflyProject/GadflyWebAPI

[21] http://GadflyProject.com

[22] https://msdn.microsoft.com/en-us/library/bb833022.aspx?f=255&MSPPError=-2147217396

One of the key drawbacks for using spaCy as our core NLP library is that it requires a significant amount of RAM space to function. Each instantiation of spaCy requires about 4GBs of memory and, therefore, requires significant capability from the server, far greater than most service providers provide for free. However, we were able to deploy our API through Heroku using a professional dyno that provides us with server capable of running the NLP code in conjunction with the web server necessary for the API. However, it must be noted that if this service was to be expanded, a better solution must be sought due to the constraints that this imposes. For example, we are only able to instantiate a single worker node and cannot handle request concurrently.

Figure 7. Homepage of the Gadfly Project Web Interface

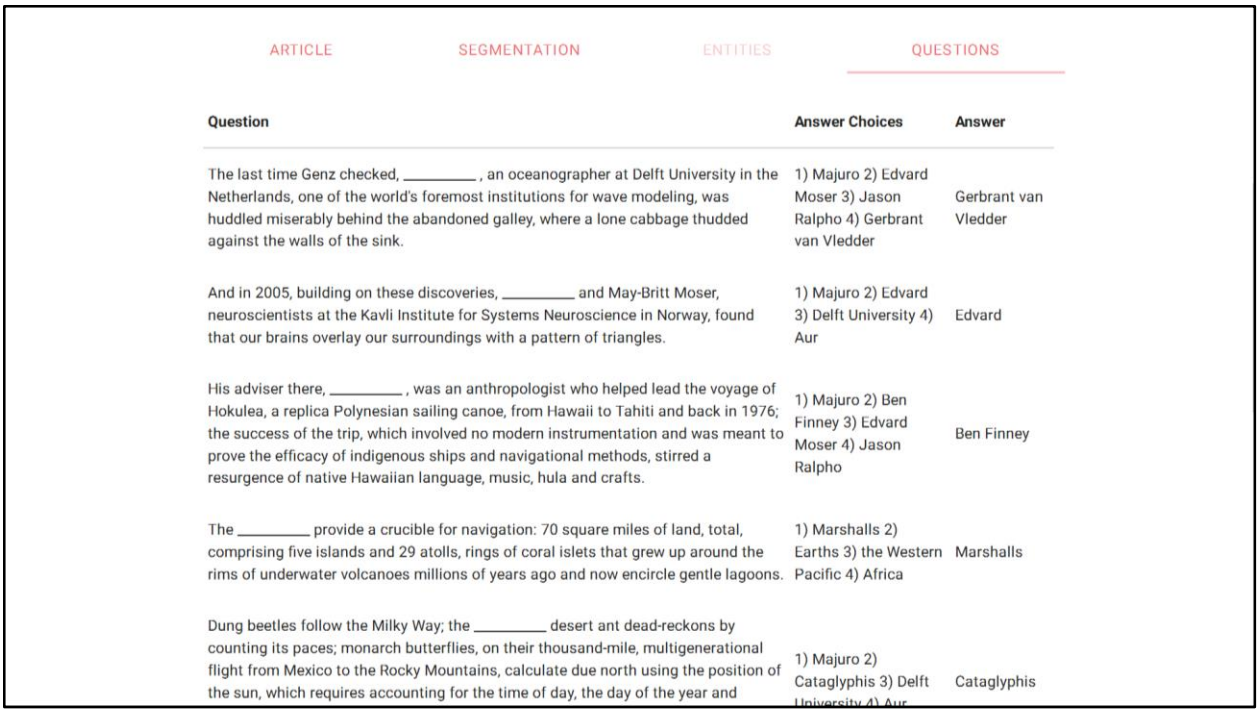


Figure 2. Automatically generated questions viewed through the Gadfly Web Interface

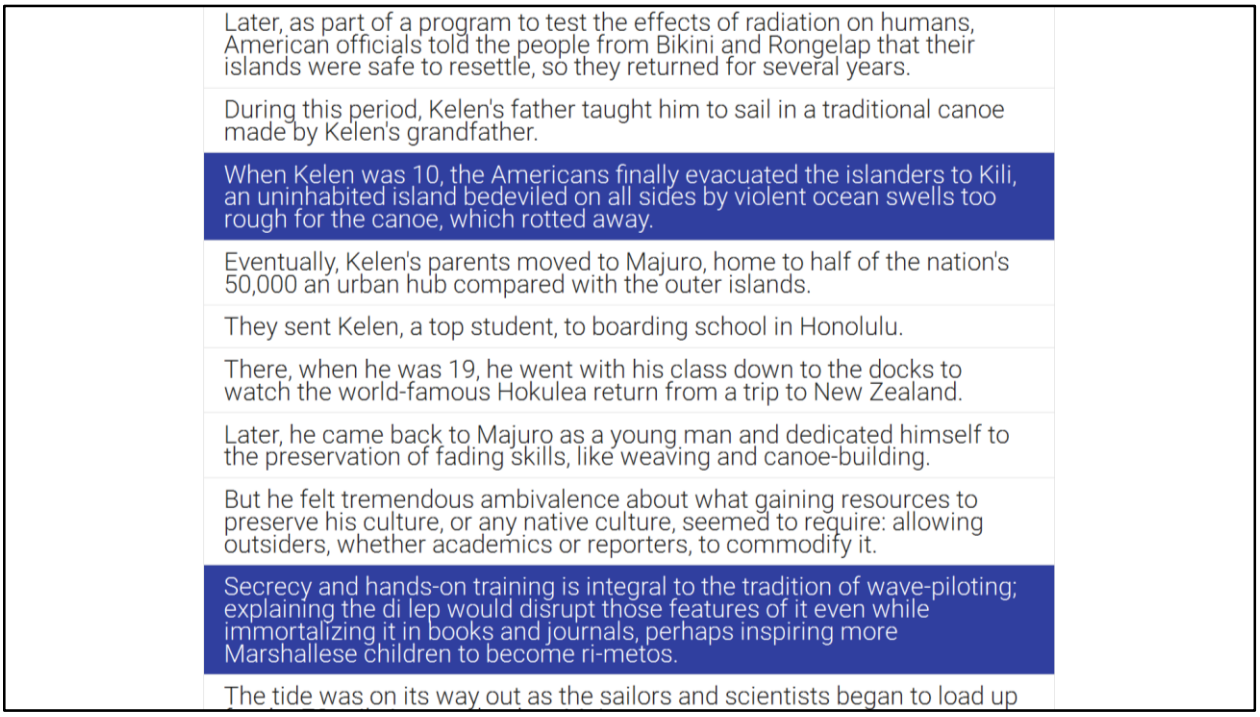


Figure 8. Top segments identified viewed through the Gadfly Project.