Generating Text through Natural Language Methods

Chirag Bharadwaj Shantanu Gore
Cornell University
{cb625,sg937}@cornell.edu

15 May 2017

Abstract

Natural language learning has been a topic of particular interest in artificial intelligence since its onset. Historical work has focused on applying classical techniques from linguistics and programming languages alike to foster the development of simple, albeit powerful, frameworks that can be used to generate text. One key challenge, though, is the *generation of unambiguous comments in English that make sense in a local context*, such as in a certain online forum.

We propose combining standard methods from the disciplines of programming languages, natural language processing, and classical artificial intelligence to develop a practical framework that can generate realistic comments in the context of forums on the social media website Reddit. This allows for simulator accounts ("bots") to generate comments around a context (e.g. as a "clean-up service" on poor-quality comments). We explore the theoretical background for this model and discuss our results as well as our evaluation of a significant implementation of such a framework.

General Terms Language, Text, Generation

Keywords Bigrams, context-free grammars, tagging, sentence structures, text generation, Reddit comments, memory-aware computing, large data analysis

1. Introduction

This project was largely motivated by the desire to apply well-studied tools from the canonical literature in a significant practical implementation. In particular, we applied standard natural language techniques and grammatical techniques from programming languages for contextual text generation. Although there are many possible use cases for such a device, we ultimately decided to use the proposed framework to generate comments that are suitable in context for Reddit forums. This paper focuses on the theory, design, and implementation of such a framework.

The rest of this paper is organized as follows. In Section 2, we document our design choices as we progressed through this project, describing initial changes to the proposal and what we eventually decided was a suitable set of goals to meet. In Section 3, we present some of the high-level details of our implementation, describing our methods of computation and any limitations that we faced while undertaking it. Section 4 features our results and examples that our model generated. Our evaluation of these results, in context, are in Section 5. Finally, our suggestions for future work can be found in Section 6. An appendix discussing the technical background necessary for the implementation of such a framework, assuming a broader computer science audience, can be found at the end of the paper.

2. Design and Structure

In designing our project, we faced many challenges and made many decisions as we progressed. This section aims to take a closer look at some of the design choices we made as the timeline for the project lapsed, as well as how closely we managed to stick to and realize our original goals.

2.1. Original Proposal

In the project proposal, submitted on 24 March 2017, we detailed the goals that we hoped to accomplish. We set out to do one major task: generate comments to be posted on Reddit. Specifically, we hoped to reply to a specific post with a suitable comment, using information both from that post and from similar posts (i.e. from other posts on this post's subreddit). We intended to use historical Reddit comments to generate these new comments. This, in fact, was one of the key reasons for choosing Reddit as our medium — it has a massive amount of historical data (on the order of terabytes) — all of which can be used to generate new comments. The theory behind how we generated comments is detailed above, in the Technical Background section, and some interesting implementation details are provided below, in the corresponding section.

We aimed to generate comments for several distinct subreddits, namely `news`, `science`, `television`, `aww` (a subreddit devoted to cute images/videos and responses to these), and `AskReddit` (a subreddit where a user posts a [generally thought-provoking] question and other users respond). These subreddits were chosen because they provide a diverse set of content types, namely technical (in the case of `news` and `television`), casual (in the case of `aww` and `television`), and a mix between the two (`AskReddit`). We separated out the subreddits by creating separate training sets for each one, and by training our models only on data from a particular subreddit. By evaluating the performance of our model on these various subreddits, we hoped to be able to gain valuable insights as to what kinds of text our model did

well on, as well what could be improved.

Originally, we intended to evaluate our system through posting comments directly on Reddit, and seeing how they performed. Specifically, we planned to use some of the metrics detailed in the Technical Background section (above), such as upvote count, total comment score, or number of gildings. We then planned to evaluate our comments based on how well they performed on these metrics, compared to other comments on the same post. In addition, we also planned to make controversial comments — in theory, these comments would not receive a significant number of upvotes or gildings, due to their controversial nature. For these comments, we intended to evaluate their "controvertiality" through the replies posted to these comments. The thought process behind this idea was that more controversial comments would get more angry replies, while more neutral comments would get more upvotes from users.

We also intended to analyze historical trends in Reddit comments, such as how language usage changes over time, or varies across different subreddits. Unfortunately, due to time constraints resulting from delays in data acquisition, we did not have sufficient time to conduct a through analysis of this topic. It remains an interesting topic for future exploration and research, as our Future Work section explores.

2.2. Revised Proposal

As our project evolved, some of our initial goals/ideas about the project shifted. Specifically, obtaining the Reddit comment dataset we needed to train on took longer than expected. An initial hurdle was actually finding where we could obtain this data. We had initially planned to simply scrape these from Reddit, but after some initial experimentation we discovered that this was easier said than done, and we did not want to spend time developing a web scraper. We also found an official Reddit API [1], but this also seemed somewhat unintuitive and cumbersome to use. After some more searching, we found a dataset that a user had generated with all of the comments from each month [2], but this came with its own issues — there was a bandwidth limit associated with downloading this dataset, and as such took several days to download.

Due to this unforeseen difficulty, we chose to focus all our effort on generating Reddit comments, rather than splitting it between that and analyzing historical trends in Reddit. In hindsight, we believe that this was the correct decision, as we were able to dive deep into this aspect of our project, and really build something significant.

Based on Professor Hirsh's feedback on our initial proposal, we also reconsidered our evaluation methods during this phase of the project. Specifically, we realized that our initial metric of simply looking at numerical scores of comments may be somewhat short-sighted, as there are a number of confounding variables that could introduce noise into these scores — popularity of the post the comment is made on, popularity of other users commenting on this post, which subreddit the comment was posted on, etc. As such, we elected to use a more focused evaluation method.

Specifically, we decided to explicitly ask volunteers (mostly other students enrolled in this course) to evaluate the Reddit account our bot was posting comments under. We decided that we would run two groups for this experiment, namely one that knew that the account which we referred them to was controlled by a bot, and another that was unaware that the comments were not generated by humans. We intended for this evaluation to supplement the numerical metrics detailed in the preceding section. Ultimately, however, we elected for a different strategy, as we discuss below.

2.3. Final Modifications

After submitting the revised proposal, we again made a couple of changes to our approach. Many of these changes stemmed from an enlightening discussion with Professor Hirsh, during his office hours on May 3rd, 2017. Specifically, after discussing our evaluation metrics, we decided that we would not actually post comments to Reddit, and solely use explicit human evaluation to judge the strengths and weaknesses of the AI we created. The reasons for this were two-fold. First of all, due to our personal usage of Reddit, we came across several bots on Reddit, and felt that they detracted from user experience. In addition, after further research, we discovered that bots were frowned upon on many of the subreddits we intended to frequent (`news`, `science`, `television`, `aww`, and `AskReddit`). Another reason for this decision was to allow ourselves to spend more time on the AI component of the project, which was our primary interest, rather than figuring out how to connect our bot to the Reddit API.

After our discussion with Professor Hirsh, and many discussions among ourselves, we came up with a few key questions that we would ask about the performance of our bot. These revolved around volunteer perception of the comments generated by the bot, based on whether the volunteer was aware that the comments were generated by a bot rather than a human. More information about this evaluation is detailed in the Results section below.

3. Implementation Details

We now discuss the actual high-level and low-level implementation details of our project. This section intends to examine some of the challenges and limitations we faced as well while progressing with our revised ideas.

3.1. Programming Languages and Libraries

We used the `python` programming language for this project, as its large collection of relevant libraries served as a huge plus in our selection. In particular, we used the following libraries when implementing our framework:

- **NLTK: Natural Language Toolkit**. For extracting parts-of-speech from English sentences. Since this is a solved problem in natural language processing, we elected to use the existing third-party `nltk` library to handle the extraction work for us.

- **Pickle**. For intermediate representations of data. Since our generative model's training dataset was on the order of terabytes (∼1.92 TiB for all comments on Reddit between December 2005 and April 2017, to be exact), we could not computationally afford to calculate results on an on-demand basis.

  Since we didn't have access to a datacenter or supercomputer, we simply spent several hours doing one-time calculations for intermediate results and ensured these were able to be integrated with our code smoothly. `pickle` helped a lot with this marshalling/unmarshalling process.

- **SimpleJSON**. For converting the Reddit dataset into Python JSON objects. The original dataset did not have the Reddit comments in as accessible of a format as we had originally hoped, so Python's built-in `simplejson` facilitated in cleaning up the dataset.

- **R.E.**. For handling regular expressions in Reddit. When parsing the JSON objects for the exact data we wanted, we obviously needed a regular expression toolkit to aid in this.

- **TQDM**. Also called *taqadum*. For printing on demand progress bars to the console. `tqdm` was extremely useful in tracking our progress when doing expensive computations (see the section on Memory-Aware Computing).

- **PipeTools**. For chaining together functions into a closure. `pipetools` offers OCaml/bash-style piping of commands into a closure, which made our final implementation at a high-level quite simple (though each constituent function was admittedly still rather complex), as in Figure 1.

It is certainly important to use the right tools for the job for a project of this magnitude, and we largely found that these libraries, toolkits, and constructs serviced our needs to the fullest extent.

```
# An arbitrary number of comments.
NUM_COMMENTS = 100

# Create a closure for comment-generation.
makeMap = (pipe
  | getData
  | convertToUTF
  | toComment
  | foreach (sanitize)
  | list
  | getBigrams
  | getRevBiMap
)

# Only have to do this once.
revMap = makeMap(files)

# Create many comments with low latency.
comments = []
for _ in range(NUM_COMMENTS):
  body = makeBiComment(revMap)
  comment = Comment(body)
  comments.append(comment)
```

Figure 1: An example of the `pipetools` package in action. This package simplified code-writing by providing functional programming constructs in Python. Our entire codebase could effectively be compressed into a single closure.

3.2. Effective Methods

3.2.1. *N*-grams

We implemented the *N*-gram generative model as mentioned in the Technical Background section as our primary infrastructure. For this, we simply parsed our custom comment objects from the large training dataset and used a random generator (Python's `random` module) with the weighted probabilities to choose how we traversed the graph of words (essentially a glorified depth-first-search).

This was fairly effective, though some other methods we attempted to impose on top of this were not as successful. For example, doing back-tracking when we failed to find a viable sentence did not help in creating more lucid sentences, despite our study, which suggested that it could be useful. The Ineffective Methods section more closely details issues with other approaches we added on top of the baseline.

We chose to use 2-grams to generate comments. We also experimented with higher values of N — including 3 and 4-grams — but due to computational difficulties, along with diminishing returns on higher values of N [8], we decided to use 2-grams. We found that this value of N provided the contextual benefits associated with N-grams (i.e. `ice cream` is more likely to be generated than `ice laptop`), while also allowing for some variability in comments. We also noted that using higher and higher values for N tends to repeat comments from the training dataset, as the set of next words becomes more limited by the words that came before it.

3.2.2. Context-Free Grammars (CFGs)

One major downside of the method of N-grams to generate text is that it pays no regard to grammatical correctness. For example, if the two sentences `That cat is so small.` and `He picked up the small ball.` appear in the training set, the sentence `That cat is so small ball.` is a perfectly valid generation. However, it is clear that this sentence is lacking grammatical sentence - specifically because it has two distinct nouns (`cat` and `ball`), without a transitive verb (such as `catched`) in between. To remedy this, we employed context free grammars.

We enforced a context-free grammar via a list of possible sentence structures (i.e. the productions, as discussed in our Technical Background section). To learn the list of possible sentence structures, however, we did *not* explicitly create a context-free grammar. Instead, we combined our model with implicit learning by using `nltk` to classify all of the sentences in the smallest subset of the training dataset (i.e.

the comments from December 2005). Specifically, for each sentence posted on Reddit in this month, we generated the list of parts of speech that it composed of. We made the simplifying assumption that all comments posted in this month were syntactically correct (followed English grammar rules). We believed that this assumption was valid because, due to the technical nature of the Reddit at that time, the comments were grammatically correct.

We then used this sampling of sentences as a list of roughly 2000 rules on which future/testing-time sentences could be generated. Although this was a compromise from the original model of providing a complete context-free English grammar, it served as a sufficient approximation to create a feasible set of comments for use in various subreddits.

After some analysis, however, we found that this method did not generate a grammar of a satisfactory quality, primarily because the grammar we had learned was somewhat complex - most sentences had lengths of around 10 words - and this led to difficulty in generating sentences. As a remedy to this, we chose several sentences that we felt could generalize well in terms of their grammar (such as `A man sat on the chair`, with part of speech tags `'determiner'`, `'noun'`, `'verb/past tense'`, `'preposition'`, `'determiner'`, `'noun'`), and used these to generate sentences instead.

3.2.3. CFG Enforcement of N-grams

We combined our N-gram based comment generation with our CFG-based structure as followed. From a set of valid sentence types (where a sentence type is a list of part of speeches), we first randomly choose a type. We then randomly pick a word which we have previously seen at the start of a sentence, then look at our bigrams map for words that could potentially follow this word, and randomly pick one based on the N-gram probabilities (more detail on this is provided in the Appendix). We then computed the part of speech tags for these two words using the `NLTK` Python library, and checked if it matched the first two POS's in the sentence type we were trying to instantiate. If not, then we restarted and chose a new first word randomly, and proceeded from there. In the case that the first two words chosen were the correct part of speeches, we repeated this process with the 2^{nd} word as the seed, this time generating a single word based on the previous one.

We also had to consider the case that there were simply no words that could follow the present word (based on N-grams), and also be of the correct sentence types. This was due to the seed word not being seen often enough in the training set, often due to a misspelling. In this case, we attempted backtracking, but in the end chose to restart the sentence generation from scratch.

Note that we did not ever check if the first seed word was of the correct type, and instead opted to generate the next word, then check if the pair of words both had the correct part of speech. This was done because POS tagging is often inaccurate for single words, due to the context-sensitive nature of POS tags. This can be seen by the fact that some words have many POS tags, and the correct tag for a word depends on the context that word is in. As such, we did not attempt to tag single words.

3.3. Ineffective Methods

3.3.1. Backtracking

In the place of random restarts, as mentioned above, we also attempted to use backtracking. In this method, if we get to a state that cannot be continued (for example, if there are no bigrams that start with a given word that also meet the desired sentence type), we remove the last word chosen and pick the last word randomly again. This has the advantage of not regenerating the whole sentence again, and preserving what was previously done. However, in practice, this method seemed to not be as beneficial as we had expected, and actually took longer to run due to spending longer in dead-end states. As such, we returned to the random restart model.

3.3.2. N-gram Trees

One difficulty we encountered was actually generating sentences using N-grams, specifically for larger values for N. The two major issues we encountered with this were space and time usage. Issues that we encountered with space usage and running time are detailed in the Memory-Aware Computing section.

We used a simple dictionary to store data needed to generate 1-grams, and a dictionary of dictionaries to store data needed to generate 2-grams. However, the implementation for generating N-grams for higher values of N was become increasingly complex, so we decided to create an abstraction to allow us to do this effectively. This abstraction was in the form of a tree. Specifically, in order to generate k-grams we created a tree of depth k, with the first $k-1$ levels used to represent the preceding $k-1$ words, and the last level used to represent the last word. Each edge in the tree would be associated with a probability of being chosen, and upon arriving at the $k-1^{th}$ a random edge would be chosen, yielding the next word. The process would then restart, taking the new $k-1$ words (including the one we just generated), and a new word would then be generated. This process would effectively continue until one of `'.'`, `'!'`, or `'?'` was encountered, signifying the end of a sentence.

The major computational roadblock we encountered with this method was simply the resources needed to run it. For example, training the model on just a couple years of data (out of the full dataset of 15 years) and trying to generate 3-grams (the lowest value of N for which this method was preferable over simple nested dictionaries) ran for four hours, then crashed due to an `OutOfMemoryError`. We predict that this was due to the way Python was representing objects in memory, and the overhead that came with that. As a result, we abandoned this approach, and decided to stick to 2-grams, for which we could use nested dictionaries.

3.4. Memory-Aware Computing

One particular point we had to be careful about when completing this project was the manner in which we handled large volumes of data. In particular, when handling the terabyes of Reddit comments, we could not afford to run our computers for hours. Notably, even a fairly powerful machine such as a 2015 Macbook Pro with an i7 processor, 16 GiB of RAM, and 512 GiB of disk space would result in the kernel killing a Python process that consumed over 50 GiB in virtual memory space:

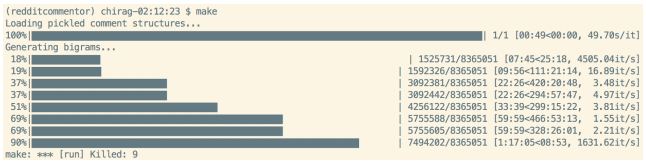

Figure 2: An example of the kernel killing an expensive Python process that was nearly finished with generation. Notably, the TQDM progress bars suggest that the kernel randomly restarts the process from a paused state as it begins to consume more and more virtual disk space (i.e. putting more pressure on the memory).

To track the source of these issues, we began monitoring all of our progress using a kernel activity monitor. We found that the disk usage was extraordinarily large, putting a lot of pressure on the memory. In particular, creating generalized N-gram trees via Python objects (as opposed to nested dictionaries) resulted in a lot of overhead, especially when we were creating on the order of billions of objects:

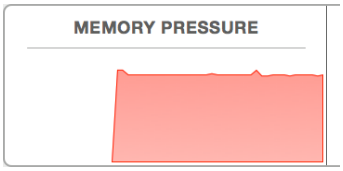

Figure 3: Our generative model putting an unbelieveable amount of pressure on the disk due to inefficient data structures.

When we switched to more effective model implementations, we found that the memory pressure eased up over time, but still consumed a lot of data at the beginning:

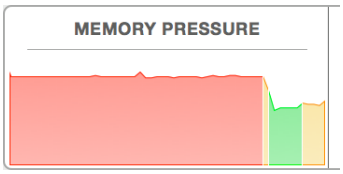

Figure 4: The generator's usage and pressurization slowly abated.

Finally, after a discussion with Professor Hirsh, we decided to be much more actively memory-conscious in our program. We began storing intermediate results (via pickling), and we were able to lower the memory constraints by a large amount as a consequence. In particular, we found that utilizing the kernel's `grep` function rather than `simplejson` cut down both the runtime and the memory utilization:

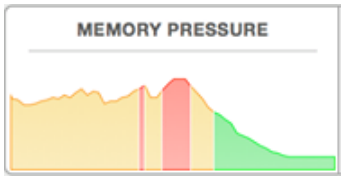

Figure 5: Using more effective methods and data structures to generate the model resulted in a cleaner experience.

3.5. Codebase

After 15 May 2017, our full codebase can be found at `https://github.com/bluedot951/redditcommentor`.

4. Results

From this generative model with a forcing function enabled on top of it, we constructed several plausible sentences in the context of the selected five subreddits. We present our results below and evaluate them in a separate section.

4.1. Early Results

Some of our early attempts at text generation utilized weaker generative models. For example, we initially used bigrams **without** a forcing function as our generator. This resulted in text such as the ones shown below:

1. `That's the popular demand, but to CNN "liberal pundits."`
2. `Paul's massive scale. C task and also invented to worry that I can't spin to the factory workers by the tribe of history, it up with. You know, a ham sandwich, is your opponent is it has confirmed that you extra $0.50 a secret executive branch predictor of the common solutions are the legislators: Penn Teller have followed by it. For several things work, where you're concerned with US companies without reason? Or in the question will leave you have any interest then surely be. And you are digital, in the work, the programming or lack jobs and point is` $x$, $y$ `such as though still stands. The article on their best at least not alowed to be careful to an option, but at a Mac Classic Maddox.`
3. `I'm "defensive" and relatively liberal media every language support this.`
4. `I think really want gem is the demo of Croquet. Totally awesome!`

4.2. Finalized Results

Our ultimate attempts (using the finalized hybrid model with generative learning in conjunction with a forced

context-free grammar approximation and sentence-length modulation) resulted in vastly different text. We present a sample of these results below as well:

1. `The cell came in that line.`
2. `The court ordered for the time.`
3. `The world were like that technology.` [sic]
4. `The cat is just :/.`
5. `The show is so paramount.`
6. `I've read novels to George.`
7. `That cat was on that lady.`
8. `This show is really somewhere.`
9. `Every generation is so criticizing.`
10. `Do you expect to know?`

Notably, the finalized sentences are much shorter (due to the length modulation parameter) and offer more in the way of semantic well-foundedness. Further scientific and experimental evaluation criteria can be found in the section below.

5. Evaluation

5.1. Survey Formulation

We had initially planned to evaluate our sentence generation model based on feedback received from comments posted on Reddit (detailed in the Original Proposal section, above), but ended up switching, for various reasons, to survey-based evaluation (detailed in the Final Modifications, above). Specifically, we evaluated our model based on feedback received our classmates. Our goal from this evaluation was to determine how effective our AI was at generating human-like comments.

We generated 100 comments for each of the five subreddits we were interested in, based on training data taken from that subreddit. From these 100 comments we then hand picked two comments to use for evaluation. This method was inspired by the methodology of Deep Drumpf, a neural network model that generates tweets based on those by Donald Trump. Specifically, this bot generates several comments, and the human user selects the best ones out of these and posts them to Twitter [15]. The 10 comments we chose, two per subreddit, are listed above (in the Results section).

In addition to these 10 comments, we also selected 10 real comments (again two per subreddit). These comments were ones that had actually been posted on reddit, and were chosen as follows. For each of the five interesting subreddits, we went to the top few posts in that subreddit from the last year. We then picked some of the top comments from these posts. We believe that the methods used for picking the two types of comments (generated and real) are similar, and as such it is valid to make comparisons between the generated and real comments. The following 10 real comments were chosen from the top all-time comments on the respective subreddits:

11. `No take, only throw.` [sic]
12. `They're like the dogs of the sea.`
13. `I thought they were too solitary for that?`
14. `Trump's style is very distinct.`
15. `Today's computers are like flipping a coin.`
16. `Except you just got baited by a decoy snail.`
17. `Just makes it easier to prove it in court.`
18. `Season 2 will have nine episodes.`
19. `Or rather they skip the previouslys.` [sic]
20. `Edit: Thanks for the gold!`

We ran two separate experiments on the 23 subjects (some of whom are listed below, in the Acknowledgements section). We split these users into four groups of 5-6 people — two control groups, and two experimental groups.

In the first experiment, volunteers in the experimental group (hereby referred to as **Group 1**) were presented with a list of twenty comments (the ten generated and the ten real ones presented above), and asked: *Select which [of the below] comments you think we were written by an AI.* In order to not provide any unnecessary influences, we noted that the number of generated and real comments may not be the same (although they were). We also created a control group for this experiment (**Group 2**), to whom we presented the exact same list of twenty comments (although in a possibly different, **random**, order), and asked: *Select which [of the below] comments you think were written by a non-native English speaker.* We provided the same statement regarding the distribution of generated and real comments to this group as we did to **Group 1**.

In this experiment, the experimental group let us ask the question *How similar are our generated comments to real user comments?*. If users thought that the comments that the AI generated were actually not AI-generated, and thus did not select them, this would mean that the comments were sufficiently coherent to be deemed as written by a human. We also included some real comments for this group, in order to see if volunteers could correctly distinguish between human written comments and AI generated comments.

The control group, in conjunction with the experimental group, let us ask the question *How does knowing that a comment may be written by an AI affect users' perception of it?* Consider a comment which users indicated as coming from a native English speaker, but also as coming from an AI. In this case, volunteers are being "harsher" on the AI than the non-native English speaker, as they are flagging the comment as "non-standard" for the AI but not for the human. This kind of analysis could also be flipped to ask a similar question.

In the second experiment, volunteers in the experimental group (hereby referred to as **Group 3**) were presented with ten pairs of comments, with both comments in each pair

coming from the same subreddit. Moreover, in every pair, one comment was generated while one was real. Volunteers were asked: *Which comment was written by an AI?*. We also created a control group for this experiment (**Group 4**), to whom we presented the exact same list of questions, and asked: *Which comment was written by a non-native English speaker?*

This experiment was similar to the previous one, with one crucial difference: participants were now forced to pick one of two comments as being from an AI, which was not required in the previous case. This distinction was made due to prior research done in the field of criminal justice, [16]. Specifically, this study found that an eyewitness who was asked to identify a criminal from a lineup of suspects (i.e. which one out of these five people is the criminal) produced a different accuracy from one who was asked if a sequence of suspects was presented (i.e. is this the criminal? How about this one? etc.). We wished to determine if this effect was also observable when picking AI-generated comments apart from human-generated comments. Specifically, conducting these two experiments simultaneously let us ask: *How does the method of presentation of data affect user behavior?*

5.2. Survey Distribution

Many of our classmates conducted similar surveys, to which they posted the links on Piazza. However, due to our fragmentation of the volunteer group, we could not simply post the link to the survey on Piazza, and instead had to ask the volunteers to send us emails. We then hand-divided the volunteers into four groups (listed above), and sent each group a separate survey. We ensured that we were putting classmates who knew what project we were doing into the experimental group, so as to not contaminate the control group (who thought that the comments were generated by non-native English speakers).

5.3. Scientific Analysis

Many interesting phenomena can be observed from the data we collected. However, it is important to first take into account confounding variables that could affect the data prior to proceeding with the analysis. In particular, it is notable that a population of 23 participants, split roughly equally across two experiments of two groups each, results in quite a small sample size for each sub-experiment. As a consequence, the statistical significance of the results must be called into question — it is entirely possible that the external variables that affect these responses may not be fully generalizable. (Nevertheless, we still proceed with an elementary analysis below.)

In particular, we must consider the impact of selecting participants that were **not** blind to our project's purpose as members of the experimental groups. Conceivably, participants that knew what they were being asked may have thought that choices were deliberately chosen to trick them, and thus respond with "counter-intuitive" choices as a reaction to the presupposed stimulus (though our survey distribution was actually quite normal with very few "trick questions", as we discussed earlier).

The results of our aforementioned survey were as follows (the order is not randomized below, but assuredly they were randomized to our participants in a double-blind fashion — we also do not know this order):

Group 1:

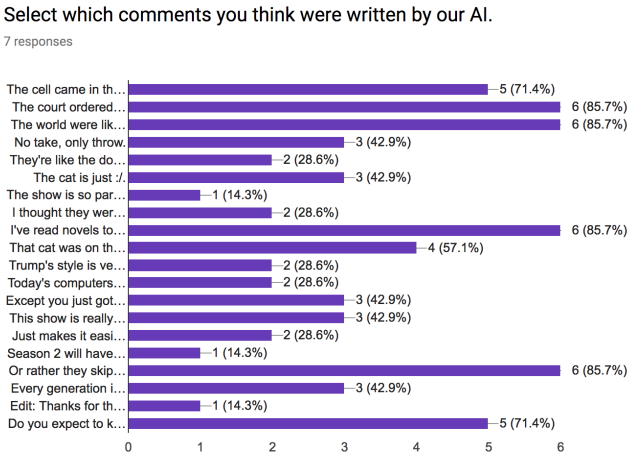

Figure 6: The experimental group for the experiment where participants were given lists of comments to select from. Comment abbreviations are visible on the left.

Group 2:

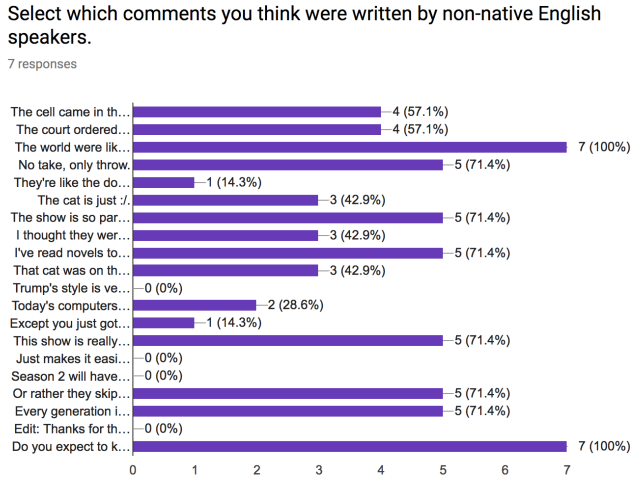

Figure 7: The control group for the experiment where participants were given lists of comments to select from. Comment abbreviations are visible on the left.

Group 3:

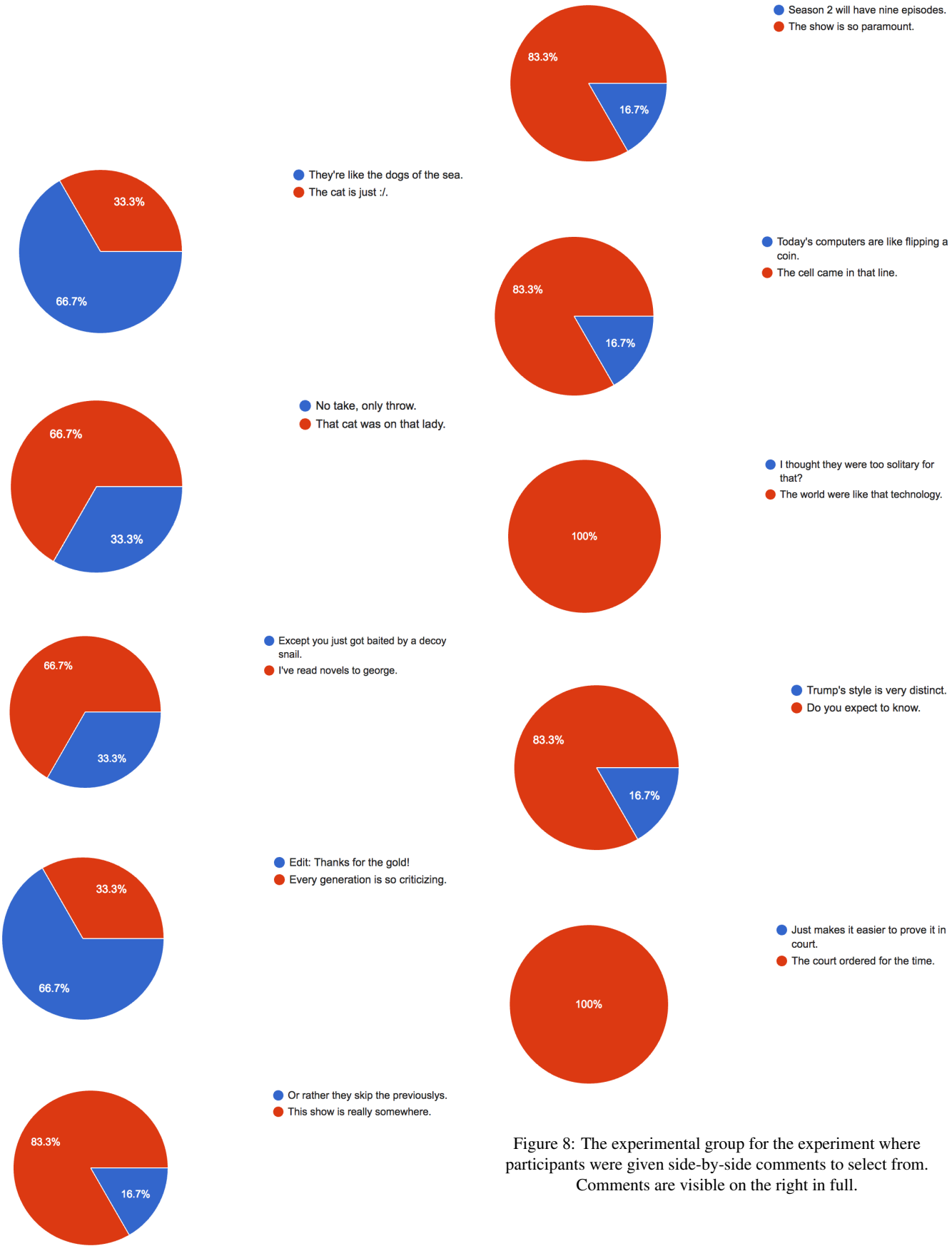

Figure 8: The experimental group for the experiment where participants were given side-by-side comments to select from. Comments are visible on the right in full.

Group 4:

Figure 9: The control group for the experiment where participants were given side-by-side comments to select from. Comments are visible on the right in full.

We now turn our attention to the fundamental questions raised in the previous section.

How similar are our generated comments to real user comments?

For this question, we focus on the data from Groups 1 and 3, since these form the experimental core of the results. In particular, we zero in on Group 1's collected data first. We perform a precision-and-recall analysis on the collected data. The precisions of the seven participants in Group 1 are:

$$\frac{9}{10} \quad \frac{8}{16} \quad \frac{9}{14} \quad \frac{4}{8} \quad \frac{7}{7} \quad \frac{4}{5} \quad \frac{3}{6}$$

Similarly, the recalls of the seven participants are:

$$\frac{9}{10} \quad \frac{8}{10} \quad \frac{9}{10} \quad \frac{4}{10} \quad \frac{7}{10} \quad \frac{4}{10} \quad \frac{3}{10}$$

Using these numbers, we can calculate the F-scores, given by $F = \dfrac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$:

$$0.90 \quad 0.62 \quad 0.75 \quad 0.44 \quad 0.82 \quad 0.53 \quad 0.38$$

On average, the F-score for Group 1 is 0.63. Note well that for Group 1, no users were actually required to select any of the comments, as the checkbox-style questions did not require any selections. For Group 3, however, we forced each participant to make a choice by providing an AI-generated comment and a real Reddit comment side-by-side, hoping that this presentation could eliminate some of the selection bias found in Group 1's results. Group 3 correctly identified comments as written by an AI around $\frac{44}{70} = 62.8\%$ of the time. (As a result, they also incorrectly identified the real Reddit comments as supposedly generated ones around $100\% - 62.8\% = 37.1\%$ of the time.)

Interestingly, these results pretty much match up with those of Group 1 (within statistical error). It is feasible to conclude that there is not a strong correlation between forcing participants to choose an answer and their process of identification. Furthermore, due to the 60+% identification rate, it seems likely that our generated comments are somewhat similar to existing ones, but there are enough internal factors to sway people to realize their nature more often than not. Reasons for this may include unusual sentence structure, repeated words, and misplaced modifiers. Also, users familiar with Reddit comments may be stronger at recognizing existing content, which bear certain hallmarks not present in the generated comments (such as "Edit: Thanks for the gold!").

How does knowing that a comment may be written by an AI affect users' perception of it?

We now take into account the behaviors of the control groups, as well. We first perform another precision-and-recall analysis on the control data. The precisions of the seven participants in Group 2 are:

$$\frac{9}{10} \quad \frac{9}{10} \quad \frac{9}{13} \quad \frac{3}{5} \quad \frac{8}{11} \quad \frac{5}{7} \quad \frac{5}{9}$$

Similarly, the recalls of the seven participants are:

$$\frac{9}{10} \quad \frac{9}{10} \quad \frac{9}{10} \quad \frac{3}{10} \quad \frac{8}{10} \quad \frac{5}{10} \quad \frac{5}{10}$$

Using these numbers, we can calculate the F-scores:

$$0.90 \quad 0.90 \quad 0.78 \quad 0.40 \quad 0.76 \quad 0.59 \quad 0.53$$

On average, the F-score for Group 2 is 0.69. Thus, comparing the F-scores for Groups 1 and 2, we see that knowing a comment may be written by an AI may result in users falsely selecting "good" (or real) comments as generated ones around 6% of the time (which is statistically significant). We perform a similar analysis for Group 4.

Group 4 correctly identified comments as written by an AI around $\frac{38}{70} = 54.3\%$ of the time. (As a result, they also incorrectly identified the real Reddit comments as supposedly generated ones around $100\% - 54.3\% = 45.7\%$ of the time.) Comparing these results with those of Group 3, we see that the control group does not know about the presence of an AI, it is much more likely to make pretty much random guesses (50-50 chances). That is, there is almost no distinction between the comments made by a native English speaker and a non-native English speaker in its eyes.

Then, looking at Group 3's data, we see that there is a statistically significant bias if a group knows that there is an AI present, as Group 3 was able to correctly identify the AI-generated comments at a rate greater than random chance. So here, we see the **opposite phenomenon** — knowing *a priori* whether a comment may be written by an AI may result in a **higher** correct classification rate (i.e. a lower false-positive rate, or a fewer number of Type I errors).

Comparing the two experiments, we see that knowing whether there is an AI present can result in different behaviors in groups of participants. However, forcing users to make a set of individual choices rather than a selection of choices improves the rate of Type I errors committed, which is consistent with the standard F-score analysis for test-accuracy measurement.

5.4. Qualitative Evaluation

5.4.1. Impact of CFGs

In this section, we examine the impact of using a CFG to force a grammar on a N-gram model. Specifically, we will look at a couple of examples constructed both with and without grammar enforcement.

Comments generated without CFGs:

- `Where is it have in?`
- `If you were just prefer.`
- `Funny how much he does.`

Comments generated with CFGs:

- `The fact is too medical.`
- `Do you feel to talk?`

- That cat is really lady.

We can make some interesting observations just from these 6 comments. First of all, the ones for which a CFG was used to enforce a grammar are significantly more coherent than when pure N-grams were used. As is discussed above, this is because the N-grams contain no grammatical information; this is added by the CFG. Note, however, that the CFG only contains syntactic information. In other words, from the perspective of a CFG, the sentences The cat sat on the chair and the sentence The chair sat on the cat are equally valid. This problem can be remedied through the use of semantics-directed generation, which is described in more detail in the Future Work section.

5.4.2. Inter-subreddit variance

In this section, we discuss some qualitative features of the comments that we thought were interesting.

First, we look at comments generated for different subreddits. We notice that for subreddits that are relatively more focused, in the sense that the comments are likely to be more similar, our AI either does very well or very poorly. This can be seen very clearly in the comments generated for the subreddit aww:

- That dog is there pet.
- Do you post to know.
- Stop them get to get.
- Do you know to post.
- Stop it make to thinks.

and the subreddit television:

- This show debuted at that attention.
- This show were that the maybe.
- This show became about the attention.
- The show is really stopped.
- Its great humor is humor.

While both of these subreddits are relatively focused (one revolves around posting cute pictures, and the other revolves around watching TV and discussing it), there is a clear superiority in the qualitiies of the generated comments for the aww subreddit than for the television subreddit. Our hypothesis is that while both these subreddits were focused, the comments made in television are textually more similar - repeat many of the same words or phrases (ie. This show) - while the comments in aww are less textually similar - different comments use a wide variety of words / grammars - while they may still be topically similar. As such, we believe that our AI performs best when provided with a wide variety of data that is textually different but topically similar

6. Future Work

Overall, our project met many of our original goals but still did not meet a few of our revised goals. To this extent, we believe that there are many directions in which this project can be extended in the future to achieve both our remaining goals as well as even greater results. Indeed, we learned many facets of the various trade-offs and setbacks in such a system. From our newfound experience, we suggest the following modifications to the baseline framework:

1. **Contractions**. Currently, our generative model strips punctuation (which is generated purely through the context-free grammar) and contractions. An interesting new direction may be to add more "casual" generated text by adding such features as contractions for well-known words. The difficulty in this is that patching existing words to their contracted varieties may not always make sense in context (for example, when not is sandwiched in between, such as in are they not? vs. aren't they?, which is literally are they not?). Furthermore, some valid English words may match with unrelated contractions (for example, wont and won't). These subtleties make handling contractions much more difficult, especially for a generative model.

2. **Predictive Modelling**. One of the problems with our current model is that words that make sense in a given local context may not make sense in a global context. As a result, pairs of words may work very well together, but the overall phrase could still be semantically meaningless. Thus, enforcing a predictive model over the CFG tree structure could help a lot by "pruning away" structures at generation-time that we know are very unlikely to occur in a grammatically-correct sentence based on the data patterns (for example, having cut-offs at each node below 5% probability of occurrence). Predictive modelling and pruning can speed up the run-time (though it poses similar issues to the back-tracking model). Having bigram tags on the actual tree at generation-time can aid in the pruning process in a practical implementation.

3. **Semantics-directed Generation**. The generative model is quite impressive in that it uses pure syntax to generate text sentences. However, ones of the drawbacks of purely syntax-directed semantics is that in reality, one requires context-sensitive grammars to answer most important questions and implement most features. However, non-local information often plays a role in context-sensitive analysis, which even the best predictive modelling may not be able to solve to the fullest extent. To aid in this, we can add *attribute grammars* [12] and parser-controlled stacks to our implementations that utilize static/dynamic semantics of a language to control generation.

   This can provide much more powerful methods of evaluation by introducing phrase structures to the model, which tag semantic meanings at generation-time. As a result, filtering upon translation from the tree to final-sentence be-

comes more predictive and more well-formed when compared to sentences that are generated from a pure branch-predictor like the previous suggestion. However, these methods come at the cost of significantly more overhead, and much more expensive computation at run-time due to more complicated parser and evaluator logic.

WordNet's `synsets` [13] provide an implementation of such a framework, which is considerably more complex than the basic generative model detailed earlier.

4. **Contextually-ambiguous Sentences**. Not all sentences that are generated can even be parsed correctly by humans. In this category, sentences generally fall into one of two categories: (i) *locally-ambiguous* sentences and (ii) *globally-ambiguous* sentences. In category (i), *garden-path sentences* often serve as a test on the efficacy of generative models for English text. For example, the sentence `the old man the boat` often trips up even native English speakers, since the most-likely semantic parsing of the sentence changes as the parser reaches the second `the` token. To sidestep the ambiguities raised in local contexts, studies [14] have proposed more complicated alternative models that use *multi-trees* with regression profiles to help eliminate the ambiguity.

   On the other hand, in category (ii), completely well-formed sentences may have multiple meanings (and thus possibly different parts of speech assigned to the same words in different meanings). This causes trouble, as a simple generative model can only generate one sentence in one way (that is, true English grammar may not result in unique parse trees, and context beyond syntactic grammar is required for complete semantic understanding). For example, the sentence `the cop chased the criminal with a fast car` exemplifies this, as the token `with` could refer to possession on either the part of the cop or on the part of the criminal, and this ambiguity, unlike the local ones from category (i), **cannot** be resolved after parsing (not even at evaluation-time) without more context. Thus, multiple-sentence-generation is required for proper creation of such sentences by a generative model, with training data on such type of sentences supplied.

7. Acknowledgements

We would like to thank Professor Haym Hirsh for the opportunity to work on such an open-ended project as well as for providing key insights when we weren't sure how to proceed with the project. His contributions were very valuable. We would also like to thank the following people for participating in surveys to aid in the evaluation of our project: Jeewon Kim, Jared Wong, Cara Zhao, Matthew Luebbers, Aleks Pesti, Drew Samuels, Timothy Chen, Lavanya Kannan, Matthew Zang, Natasha Armbrust, Smit Jain, Zachary Bamberger, Sonia Appasamy, Grant Gonyer, Nicole Pirringer, XuTing Tao, Abhimanyu Gupta, Jeevan Karamsetty, Arshi Bhatnagar, Alan Zhang, Andrew McHugh, Daniel Li, Ahaan Nachane, and other anonymous reviewers.

References

[1] Alexis Ohanian and Steve Huffman. *Reddit API Documentation*. `https://www.reddit.com/dev/api`. 2005.

[2] `/u/Stuck_in_the_Matrix`. *Reddit Comments Directory*. `http://files.pushshift.io/reddit/comments`. 2014.

[3] Edward Snowden, Laura Poitras, Glenn Greenwald. "Ask Me Anything!". `https://redd.it/2wwdep`. 2015.

[4] Claire Cardie. *Lectures in Natural Language Processing*. `https://www.dropbox.com/sh/z5r2qaogzv2hj2a/AAATOGfJdb4MjHvsBbq8-GTsa`. Lectures 2-4 (*N*-grams). 2015.

[5] Chris Harrison. *Visualizing Google's Public Tri-gram Data*. `http://chrisharrison.net/index.php/visualizations/webtrigrams`. 2006.

[6] Daniel Kudenko and Haym Hirsh. *Feature Generation for Sequence Categorization*, in AAAI Proceedings. Rutgers University: Piscataway, NJ. 1998.

[7] Anton Antonov. *Markov Chains N-gram Model Implementation*. Wolfram Research, Inc.: Coral Springs, FL. 2014.

[8] Ethan Miller, Dan Shen, Junli Liu, and Charles Nicholas. *Performance and Scalability of a Large-scale N-gram-based Information Retrieval System*, in Journal of Digital Information. University of Maryland, Baltimore County: Baltimore, MD. 2000.

[9] Dexter Kozen. *Automata and Computability*. Springer-Verlag: Ithaca, NY. 1997.

[10] Noam Chomsky. *Three Models for the Description of Language*, in Journal of Symbolic Logic. Massachusetts Institute of Technology: Cambridge, MA. 1956.

[11] James Higginbotham. *English is not a Context-Free Language*, in Studies in Linguistics and Philosophy. Massachusetts Institute of Technology: Cambridge, MA. 1984.

[12] Arthur Pyster. *Semantic-Syntax-Directed Translation*, in Journal of Computer and System Sciences. Ohio State University: Columbus, OH. 1978.

[13] George A. Miller. *WordNet: A Lexical Database for English*. `https://wordnet.princeton.edu`. Princeton University: Princeton, NJ. 1985.

[14] Weijia Ni, Stephen Crain, and Donald Shankweiler. *Sidestepping Garden Paths: Assessing the Contributions of Syntax, Semantics, and Plausibility in Resolving Ambiguities*, in Conference on Sentence Processing. Yale University: New Haven, CT. 1994.

[15] Brad Hayes. *Deep Drumpf: A Neural Network Trained on Donald Trump*. `http://www.deepdrumpf2016.com`. 2016.

[16] Beth Schuster. *Police Lineups: Making Eyewitness Identification More Reliable* in NIJ Journal. Office of Justice Programs: Washington D.C. 2007.

Appendix: Technical Background

In this appendix, we present the technical background necessary to understand the rest of the paper. In particular, we discuss details of the Reddit social media site, the concept of N-grams, and the mathematical precepts of context-free grammars in a broad context.

7.1. Reddit

Reddit is an American social media site in which users can participate in a forum-based discussion. Participating users must have a username, but no other private details (such as location, age, etc.) are divulged on a user's profile, unlike in other online forums. On the other hand, Reddit does not afford true anonymity to its participants (unlike on 4chan, a similar and older online forum). A point of note is that as Reddit grew in popularity over the last several years, it was instrumental in broadcasting a wide variety of social, political, and economic events. Furthermore, Reddit is also used as a platform for teaching, learning, and controversial discussion. Each of these motives play a role in the types of comments found on the various forums.

Participants ("users") on Reddit can post comments, upvote/downvote others' comments (i.e. share their appreciation or distaste anonymously, respectively), and donate to users ("gild") whose comments were particularly witty or memorable. The "score" of a Reddit comment is the collective sum of its upvotes and downvotes ($+1$ for upvotes and -1 for downvotes, respectively). A "gilded" comment has a gold star next to it, but does not share the name of the donor. Figure 6 shows an example of a discussion on Reddit with these metrics displayed.

Reddit has a public API [1] from which data can be scraped. In particular, all of the comments on all of the threads ("posts") on all of the forums ("subreddits") are available for extraction. This data is complete in that it dates back to Reddit's inception in late 2005. In mid-2014, an online database [2] was created by querying the API for a complete collection of posts and comments since Reddit's creation. The collection is currently batch-updated at the end of each month with the complete archived comments from that month. We sourced all of our training data by sampling this.

7.2. N-grams

One of the techniques we employed came from the subdiscipline of natural language processing. In particular, we used the studied technique of N-grams [4] to help in modelling a subset of the English language. We detail some of the important aspects of this modelling procedure below.

A common approach to language modelling is to use *generative models*, which explicitly describe a method (or methods) by which text in a language can be generated. In other words, generative models offer a sense of word-prediction in a given context (hence their attractiveness for use in the aforementioned project).

Language models pose a significant advantage in today's world in that they can serve as the backbone for several

Figure 10: An example of comments, upvote/downvote ratios, and gildings on a Reddit thread in which Edward Snowden was active in 2015 [3], demonstrating Reddit's sociopolitical value.

services. To name a few, augmented communication systems (e.g. for the disabled or elderly) and automated user-stimulated response systems (e.g. so-called "chat-bots" for technical support) certainly are aided significantly by a generative English model in their core infrastructures. Our project most closely mirrors the latter use case inasmuch as we use such a model to generate contextually-correct comments in a given subreddit.

The technique of *N-grams* is a generative model that assigns probabilities to consecutive sequences. For example, we consider a lexicon L of n words:

$$L = \{w_1, \ldots, w_n\}.$$

Generative learning uses the probability of an upcoming word given the words that have already occurred to make predictions. For example, if we have already generated the sequence of two words $W = w_i, w_j$ (for $1 \leq i, j \leq n$), then we can ascertain the probability that some w_k (again for $1 \leq k \leq n$) is generated next as a conditional probability on the known information:

$$P(w_k | w_i, w_j).$$

(Where this probability is between 0 and 1.) The generative model, as its namesake suggests, generates these probabilities and uses them to select which words to come next in a sequence. The generative model can achieve this by using a classic two-step process from machine learning:

- Training-time sampling of existing real data.
- Testing-time generation of plausible new data.

At *training-time*, the generative model uses large volumes of real data to compute the various probabilities of adjacent

occurrences. In particular, the N-grams model considers sequences of N adjacent words at a time to compute the various probabilities using **relative** frequency counts in a map of words to probabilities. Then, at *testing-time*, the model initially generates the first word in a *sentence* by randomly choosing a seed word from a set of "most likely sentence-starters". Thereafter, the generative model utilizes its generated conditional probabilities map to randomly choose the next words in the sentence, adding more and more known/-given information in its conditional probabilities with each additional *epoch* (i.e. each step at which another word added to the generated sentence).

A visual depiction of this model can be found in a graph form (where vertices are words and edges are probabilities) in Figure 11. Note that the model does not prevent words from being generated multiple times in a sentence, since it simply wraps a stochastic model (i.e. an order $n-1$ Markov chain) over existing data to aid in the generation of new data.

As Figure 11 suggests, for the case of a 2-gram generative model (*bigrams*), the third word generated in a sentence does not **directly** depend on the first word generated, as there may be multiple conditional paths from a seeded start word to that word. Since certain words in the training data set may have ending-punctuation identifier after it, a simple model can parse these out as additional "words" and simply choose to end a sentence if it happens to generate such a keyword along the graph. Intuitively, the *non-parametric* N-grams model scales well (in accuracy) with N due to the fact that the generative model for $(k+1)$-grams includes the generative model for k-grams as a subgraph. Indeed, studies [6, 8] confirm that this model produces increasingly more realistic comments as N becomes large. Table 1 on the following page also demonstrates this phenomenon in practice when a large lexicon of words is provided as input to the model.

It is precisely this flexibility and power that the non-parametric, generative N-grams model affords that made it an extremely important part of the core design of a system for our project. However, the model can be vastly improved by using other techniques to actually explicitly enforce the English grammatical structure over any generated text, since the N-grams model simply implicitly learns it from what it sees in the large training dataset. To this extent, we also consider the idea of explicit grammatical structure as a forcing function over generated text.

7.3. Context-free Grammars

Another technique that we employed came from programming language theory. As we mentioned earlier, the enforcement of a grammar over the generated text could serve as a useful tool to aid in the creation of meaningful sentences. The reasons that N-grams alone provide a limited model for generation will become apparent momentarily, but the technique of grammar-based generation aids in driving meaningful text.

In his 1956 seminal paper, Noam Chomsky introduced the notion of what are now known as context-free grammars as cogent (albeit powerful) descriptors of modern language [10]. Kozen formulated a clear description [9] of context-free grammars in his text for Cornell undergraduate students; we present Kozen's recapitulation of Chomsky's original work below. □

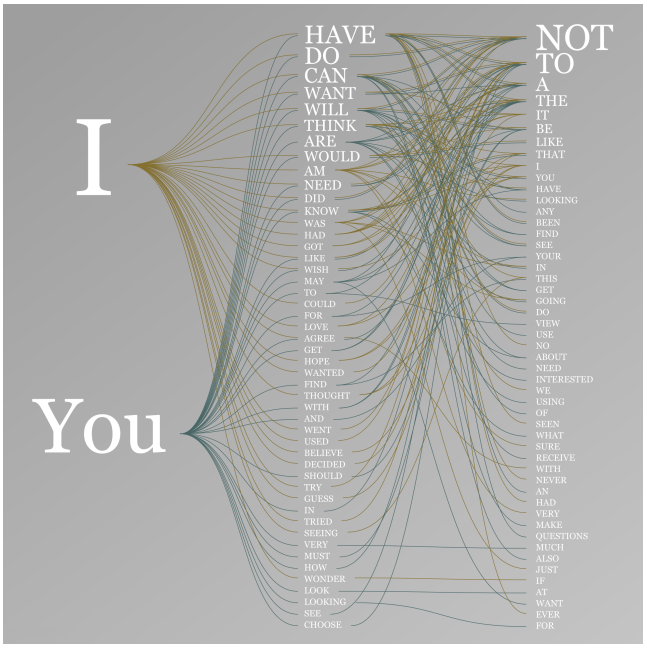

Figure 11: A graph visualization [5] of bigrams.
Notice that the graph is not necessarily complete at each level.

A *context-free grammar* (CFG) is a finite set of *rules* defining the set of well-formed syntactic expressions in a language. Formally, a CFG is a quadruple

$$G = (N, \Sigma, P, S),$$

where:

- N is a finite set (the *nonterminal* symbols),
- Σ is a finite set (the *terminal* symbols),
- P is a finite subset of $N \times (N \cup \Sigma)^*$ (the *rules* or *productions*), and
- $S \in N$ (the *start symbol*).

We use capital letters A, B, C, ... for nonterminals and lowercase letters a, b, c, ... for terminal symbols. Strings in $(N \cup \Sigma)^*$ are denoted with Greek letters α, β, γ, ... Furthermore, instead of writing productions as (A, α), we introduce the shorthand notation: $A \to \alpha$. We also often use the vertical bar | to abbreviate a set of productions with the same left-hand side. For example, instead of writing three similar rules separately as

$$A \to \alpha_1 \qquad A \to \alpha_2 \qquad A \to \alpha_3,$$

we can combine them into a single production using |:

$$A \to \alpha_1 \mid \alpha_2 \mid \alpha_3.$$

N	**Generated Text**
2	only by taking the case with those from very distinct species; for if on the old spanish pointer came from spain, mr. borrow has not in this predicament; for she makes her own offspring. the occasional influence of intercrosses, even at this rate there must have been watched, and tended the larvae, though active, still obey more or less obscured, of the multiple origin of our palaeontological collections. that our knowledge, imperfect though it must not be identically the same; but they refuse to sum up, i believe that such modifications are accumulated through natural selection; organs of fishes offer another case of most naturalists, the structure of various regions. we see them, well defined? secondly, is it surprising that the good swimmers if they pass through a lens, but not quite regularly, and producing new species. that natural selection in their offspring is notorious; but some taken out of the facts briefly given in this strange habit, and partly methodical selection. perhaps the first chapter, that there are, which, though probably formed at the fourteen-thousandth generation, will probably be much reduced or rendered rudimentary at this rate there must be slow and scarcely sensible mutation of the sea. on the
3	only by giving long catalogues of facts. we shall, however, be enabled to trace in an admirable manner the former migrations of the inhabitants within each great class, generally change at a slower rate than the intermediate varieties, which exist in lesser numbers; so that the islands are situated on moderately deep submarine banks, and they are not supposed all to appear simultaneously, but often after long intervals of time; for fossiliferous formations, thick enough to resist such degradation as it has been studied; yet a german author makes more than a truism, for if a variety were to flourish so as to perform all the work by itself, being aided during the process of selection, notwithstanding a large amount of inheritable and diversified variability is favourable, but i believe mere individual differences suffice for the work. a large number of individuals of many species having similar habits with the rock-pigeon seems to me certainly to hold good when first one variety and then several mixed varieties of wheat have been sown on equal spaces of ground. hence, if any one species of swallow having caused the decrease of another species. the recent increase of the missel-thrush in parts of the sea will
4	only by giving long catalogues of facts. we shall, however, be enabled to discuss what circumstances are most favourable to variation. in the next chapter i shall discuss the complex and little known laws governing variation are the same, as far as we can see, with the laws which have governed the production of so-called specific forms. in both cases physical conditions seem to have produced some effect; for it is difficult to avoid believing that they are closely consecutive. but we know, for instance, from sir r. murchison's great work on russia, what wide gaps there are in that country between the superimposed formations; so it is in north america, and in many other animals, and in flowers, that organs, which when mature become extremely different, are at an early stage of growth exactly alike. how inexplicable are these facts on the ordinary view of independent creation; whereas on the view here given in regard to the later embryonic stages of our domestic varieties. fanciers select their horses, dogs, and pigeons, for breeding, when they are nearly grown up: they are indifferent whether the desired qualities and structures have been acquired earlier or later in north america than in those of europe; time
5	only by giving long catalogues of facts. we shall, however, be enabled to discuss what circumstances are most favourable to variation. in the next chapter the struggle for existence amongst all organic beings throughout the world, which inevitably follows from their high geometrical powers of increase, will be treated of. this is the doctrine of malthus, applied to the whole animal and vegetable kingdoms; for in this case there can be no artificial increase of food, and no prudential restraint from marriage. although some species may be now increasing, more or less rapidly, in numbers, all cannot do so, for the world would not hold them, the more dominant groups beat the less dominant. this tendency in the large groups to go on increasing in size; and they consequently supplant many smaller and feebler groups. thus we can account for the fact that all organisms, recent and extinct, are included under a few great orders, under still fewer classes, and all in one great natural system. as showing how few the higher groups are in number, and how widely spread they are throughout the world, the fact is striking, that the discovery of australia has not added a single insect belonging to a new order;

Table 1: Generated text by an N-grams model [7] for various values of N. The training data set was sampled from Charles Darwin's "The Origin of Species" (1859). The semantic comprehensibility of the generated sentences visibly increases on an exponential scale as N grows larger. The original text has 215,163 valid, unique words, based on a computer program's parsed output.

Furthermore, for two strings $\alpha, \beta \in (N \cup \Sigma)^*$, we say that β *is derivable from* α *in one step* if β can be obtained from α by replacing some occurrence of a nonterminal symbol A in α with some string γ, where $A \to \gamma$ is in P. That is, if there exist $\alpha_1, \alpha_2 \in (N \cup \Sigma)^*$ and a rule $A \to \gamma$ such that

$$\alpha = \alpha_1 A \alpha_2 \quad \text{and} \quad \beta = \alpha_1 \gamma \alpha_2.$$

We mathematically denote that β is derivable from α in one step (over the grammar G) as the relation

$$\alpha \xrightarrow[G]{1} \beta.$$

We also define the reflexive-transitive closure $\xrightarrow[G]{*}$ of the relation $\xrightarrow[G]{1}$ using the following inductive procedure:

- $\alpha \xrightarrow[G]{0} \alpha$,
- $\alpha \xrightarrow[G]{n+1} \beta$ if $\alpha \xrightarrow[G]{n} \gamma \xrightarrow[G]{1} \beta$ for some γ, and
- $\alpha \xrightarrow[G]{*} \beta$ if $\alpha \xrightarrow[G]{n} \beta$ for some $n \geq 0$.

If $S \xrightarrow[G]{*} \alpha$, where $\alpha \in (N \cup \Sigma)^*$, then α is called a *sentential form*. A sentential form is called a *sentence* if it consists of **only** terminal symbols. Thus, a sentence is an element of Σ^*.

The *language generated by* such a context-free grammar G, denoted by $L(G)$, is the set of all possible sentences:

$$L(G) = \{x \in \Sigma^* \mid S \xrightarrow[G]{*} x\}.$$

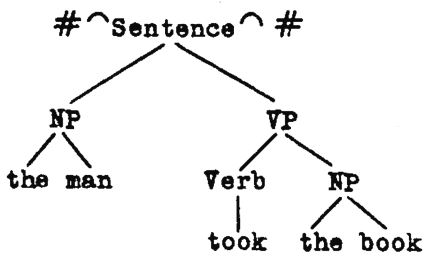

Figure 13: A parse tree for "`the man took the book`".

A subset of sentences $B \subseteq \Sigma^*$ is called a *context-free language* if there exists some backing grammar G for which $B = L(G)$. □

Although studies have shown that English **cannot** be represented by a context-free grammar [11], they have also suggested that it is largely possible to closely approximate a subset of it using appropriate features and such generative techniques as the N-grams model. One of our project's goals is to achieve this aspect of generation by using a context-free grammar to aid in the process.

One can realize an approximate context-free grammar for English using techniques that Chomsky explored in his original work in conjunction with the aforementioned mathematical descriptions. In particular, context-free languages can be split up as parse trees, commonly represented as *sentence diagrams* in grade-school English classes. For example, let us consider the following small subset of English, represented as a context-free grammar in Figure 12:

$$
\begin{aligned}
S &::= NP\ \ VP \\
VP &::= V\ \ NP \\
NP &::= \texttt{the man} \mid \texttt{the book} \\
V &::= \texttt{took}
\end{aligned}
$$

Figure 12: A simple context-free grammar for a subset of English with only four possible generatable sentences.

While this is a limited subset, we can still construct interesting examples. Let us consider the following "sentence" (in the sense of context-free grammars):

`the man took the book.`

We can then construct the sentence diagram shown in Figure 13 (i.e. parse tree in the sense of CFGs):

This type of structure is actually **quite** easy to implement in practice, and as a result serves as a useful forcing function when generating text. The tree-like structure that a context-free grammar imposes allows implementers to freely utilize language-level constructs to internalize the grammatical structure of a natural language, thereby allowing one to perform, say, a tree traversal at generation-time when using a method such as N-grams. Indeed, our project implementation ultimately ended up following this procedure, as the Design/Structure and Implementation sections suggest.