Investigation into Grading English Grammar

Martha Bellows   Antoinette Bongiorno   Brandon DiGiulio

May 12, 2017

1 Abstract

Despite advancements in Abstractive and Compressive summarization, the community standard for scoring automatic summarization neglects to objectively address grammaticality. In this paper we attempt to present an objective and replicable solution to this concern. Using the NLTK and the Penn Treebank Tagset, we generate an underlying rule set of POS tag sequences from a combination of modern and classic literature, assuming these widely accepted texts have correct grammar. This underlying rule set is used to lookup and match POS tag sequences, generated from input sentences. While further exploration is need to ensure the underlying rule set is optimally robust, our experimentation thus far has made visible progress towards an objective solution to grammaticality scoring of automatic summarization outputs.

2 Introduction

In the field of Automatic Text Summarization, there are two main categories of research: extractive and abstractive summarization. Extractive summarization describes the process of selecting contextually important sentences from the document and concatenating them to form a summary, while abstractive summarization aims to generate novel sentences that represent the main idea in a more concise manner. With ad-vances in natural language processing and computer science, abstractive summarization and its newer counterpart, compressive summarization, have grown in popularity. Despite this growth in the field, it seems the methods used to evaluate progress have not scaled well, causing the community to continue to rely on human subjectivity as a metric in grading grammaticality and linguistics. In this paper we propose a model for evaluating grammaticality in a more objective and replicable way.

2.1 Motivation

In recent years, there has been additional research in compressive summarization with a focus on grammar. In a paper by Li, Liu, Liu, Zhao, and Weng, we see the author's claim to make significant improvements in the linguistic quality of generated sentences using ILP and constituent parse trees. [1] Curious to realize just how significant the improvements were, we looked into their method of measurement. In the paper, Li et al. cites that they "invited [three] native English speakers to do the evaluation" [1]. Taking note of the lack of replicability of this method, we recognized that there is a need for a more objective way to score the grammaticality and linguistic quality of sentences in order to further research in the growing field of Automatic Text Summarization. A more consistent method of grammaticality evaluation will help researchers measure the effective-

ness of their systems, ranking them against competing research.

2.2 NLP Community

As research began with extractive summarization, it seems that the focus of Automatic Text Summarization is still on content and salience. In extractive summarization, the sentences that comprise the summary have not been altered in a way that could change their grammar, and assuming that the source document has correct grammar, there is not a need to score grammaticality. Due to this, the community standard for grading automatic summarization is the ROUGE metric [2]. This metric uses a comparison of n-grams between the summary to be evaluated and one or several human-written reference summaries, essentially scoring the content of the summary. Yao, Wan, and Xiao write that aside from content, evaluation on other aspects of summaries relies heavily on human judgement. At the Document Understanding Conference (DUC) which later became known as the Text Analysis Conference (TAC), the main conferences in the field, human judges are asked to rate different aspects of the summaries. This manual evaluation, on grammaticality, non-redundancy, clarity, and coherence, is said to be indispensable as there is no better solution at this time [2].

This method of evaluation is used across the community. We see Li et al. asking a small sample of English native speakers to give a rating, on a scale from 1, bad, to 5, good, for both grammar and coherence [1]. Woodsend and Lapata conduct their human judgement evaluations over the internet using Mechanical Turk, to generate a larger sample of 54 volunteers. These self-reported native English speakers grade each summary for grammaticality and informativeness, on a scale of 1 to 5, after successful completion of comprehension questions [3]. In our research, we are curious as to why a better, more replicable and less subjective solution has not been developed, or been applied, before. At this time, there are commercial solutions that grade grammar, but none have been applied to this community that we saw in our research. In the model presented in this paper, we attempt to find a way to bridge this gap, so that automatic summarization research does not have to rely on the subjectivity of human judgement.

2.3 Grammar

In order to grade a summary on grammaticality, a definition for grammar needs to be established. Grammar is, simply put, a system of rules and syntax that defines how things should be written and spoken. Grammar gives communication an understood, defined meaning between two or more parties. These rules can be further defined using words' parts of speech and their relation to each other. "We say a sentence is grammatically well formed when the adjacent words

are in agreement with each other. That is, the parts-of-speech (POS) tags of adjacent words are mutually compatible, where the level of compatibility accounts to the degree of acceptance" [4]. Using this, we can determine the importance of compatible combinations of POS tags in grading overall grammar.

3 Parts of Speech Tagging

Our idea is to develop a way to score the grammatical correctness of an input sentence based on the comparison of that sentence's POS tag sequence to a generated grammar rule set. We are focused only on the POS tag sequences because as stated by Jurafsky and Martin, POS tags are significant for the "large amount of information they give about a word and its neighbors" [5]. The POS tag ruleset can be produced by processing a large number of grammatically correct sentences and analyzing these correct combinations of respective POS tags. By similarly processing automated text summaries into series of POS tags, we have the foundation for comparing a known set of correct sequences to these new, grammatically ungraded sentences. This grammaticality scoring method could someday replace human judgement in Automatic Text Summarization evaluation which will make this grading process more objective, reliable, and repeatable.

3.1 Natural Language Toolkit

Tag	Description
$	Dollar sign
"	Quotation mark
(	Opening parenthesis
)	Closing parenthesis
,	Comma
–	Dash
.	Sentence terminator
:	Colon or ellipsis
CC	Conjunction, coordinating
CD	Numeral, cardinal
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or conjunction, subordinating
JJ	Adjective or numeral, ordinal
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal auxiliary
NN	Noun, common, singular or mass
NNS	Noun, plural
NNP	Noun, proper, singular
NNPS	Noun, proper, plural
PDT	Pre-determiner
POS	Genitive marker
PRP	Pronoun, personal
PRP$	Pronoun, possessive
RB	Adverb
RBR	Adverb, comparative
RBS	adverb , superlative
RP	Particle
SYM	Symbol
TO	"To" as preposition or infinitive marker
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, present participle or gerund
VBN	Verb, past participle
VBP	Verb, present tense, not 3rd person singular
VBZ	Verb, present tense, 3rd person singular
WDT	WH-determiner
WP	WH-pronoun
WP$	WH-pronoun, possessive
WRB	WH-adverb

Figure 1: Penn Treebank Tagset

In order to do the work of tagging sentences with POS tags, we utilized the Natural Language Toolkit (NLTK) [6]. NLTK is written in python and has many functions for natural language processing. In particular, we used NLTK in order to tokenize and tag sentences to build a text file of POS tag sequences. The tags used by NLTK is the Penn Treebank Tagset (Figure 1). The Penn Treebank Tagset was created from the origi-

nal 87-tag tagset for the Brown corpus [5].

Other systems were considered such as RASP and an NLP system created by Stanford. RASP (Robust Accurate Statistical Parsing) is a system designed for syntactic annotation of free text [7]. It is implemented using C and Common Lisp and uses POS tags derived from the CLAWS tagset. The CLAWS tagset is comprised of over 130 tags, depending on the version, so it is much more extensive than the Penn Treebank Tagset. For example, the CLAWS tagset goes into finer detail with verbs specifying individual tags for the verbs to be, to do, and to have. It even goes so far as to break out each of these verbs to have a tag per verb tense. Because RASP uses a larger tagset, we deemed it would be beneficial to start with the simplest tagset available.

The other system we considered is the NLP software available from The Stanford NLP group. Like NLTK, it uses the Penn Treebank tagset but instead of being implemented in python, they use Java. Ultimately, we decided to use python for our initial investigation because it is a good starting point. In addition, the documentation available from NLTK is extensive and easy to read.

3.2 NLTK - Tagging Sentences

In using NLTK, we are able to transform regular sentences into POS tags. There are four stages in this process from getting it from the initial sentence to a sentence with POS tags. Starting with initial sentences (1), it is necessary to convert it so each sentence is broken down into its individual sentences (2). This process is known as tokenizing sentences. The next step is to break each sentence into its individual words and punctuation which is called word tokenization (3). The final step is to tag each word in the sentence (4).

1. Initial Sentence: 'I went on a walk. It was nice outside.'

2. Tokenize Sentence: ['I went on a walk.', 'It was nice outside.']

3. Tokenize Words: [['I', 'went', 'on', 'a', 'walk', '.'], ['It', 'was', 'nice', 'outside', '.']]

4. Tag Words in Sentence: [('I', 'PRP'), ('went', 'VBD'), ('on', 'IN'), ('a', 'DT'), ('walk', 'NN'), ('.', '.')]

These steps and the overall algorithm are further defined in Section 4.

3.3 Project Gutenberg

In order to build the rule set of correct tag sequences, literature from Project Gutenberg [8] and books from our personal electronic collections were used. Project Gutenberg has a collection of free electronic books totaling more than 53,000. The majority of them are free because the books are considered public domain. A list of books used to create our file

of tag sequences can be found in Appendix A.

We used standard UTF-8 encoded text files for the books to use in our script for creating tag sequences. For Project Gutenberg, we individually downloaded the .txt version. For our personal books, we converted them from .mobi to .txt. The conversion of .mobi to .txt was done using the program Calibre. All files were then manually preprocessed and stripped of any text preceding or coming after the main text of the book. This includes title page information, table of contents, letters by the author, acknowledgments, glossaries, etc.

Initially, we started only using four pieces of literature [Appendix A: 1, 4, 5, 9]. They were picked based on their popularity and their availability at Project Gutenberg. Because of their world-wide renown and the fact that they are published professionally, proper grammar is assumed. Eventually we had to expand our data set so we added ten more books from Project Gutenberg [Appendix A: 2, 3, 6, 7, 8, 10, 11, 12, 13, 14]. This was still not sufficient to what we were trying to accomplish and reasoned that we should start including more recent pieces of literature. Up to this point, the most recent publication we had was from 1911 [Appendix A: 11]. Since recent literature is not available on Project Gutenberg, we started using personally purchased literature [Appendix A: 15 - 70].

All of the books in Appendix A were sequentially run through the algorithm described in Section 4.1: Generating Rule Set.

4 Algorithm

4.1 Generating Rule Set

In this section, we go over how our algorithm generates its underlying rule set. Based on existing, assumed grammatically correct, pieces of work [8], we build a pseudo-database of correctly formed grammar combinations. Books are first read into memory by the application, one at a time. Using the Python NLTK library, each book is tokenized into separate sentences and then further tokenized into individual words per sentence. Each sentence is then treated separately, as each word contained within is tagged with the appropriate Part of Speech (POS) tag generated by the NLTK library. The result is a list of tuples each containing a word of the sentence and the POS tag. Since our algorithm will only care about the POS tags to check grammar, the original word is removed and all that remains is the sequence of POS tags for each sentence.

Next the algorithm compares these new sequences with the current database to check for any existing matches. A binary search is done (the database is kept sorted) and if a match occurs, it will not be inserted so we avoid duplicated sequences. If there are no matches found at the end of the binary search, the sequence is added and the list

sorted. Through this part of the overall algorithm, a ruleset to compare summaries and other sentences has been created.
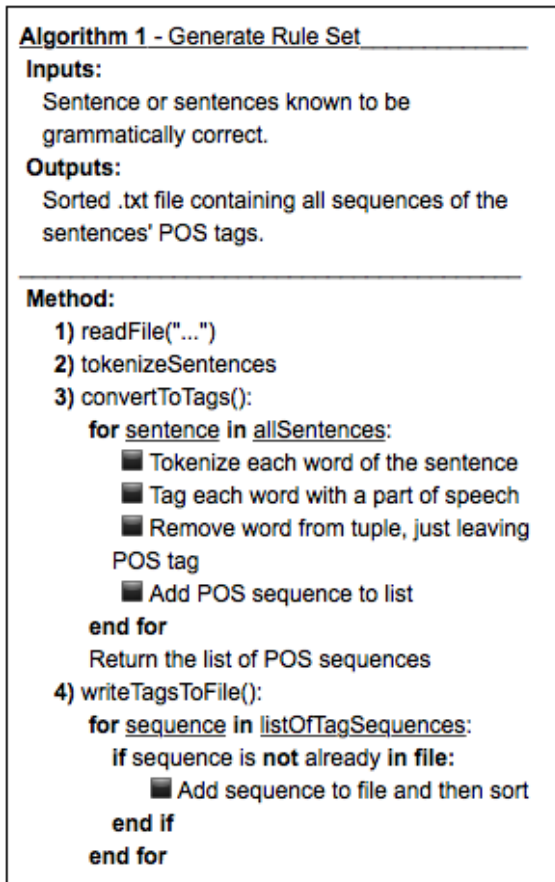

Figure 2: Algorithm 1 - Generating Rule Set

4.2 Grading Against Rule Set

In order to grade now against the aforementioned ruleset, we must now compare the grammar sequences of sentences to be graded with the ones that we already know to be correct. The new sentences must first be handled as the ones were in Section 4.1. They are first read and then processed using the NLTK library. Again, this tokenizes them into individual sentences, then words, and finally broken into sequences of POS tags.
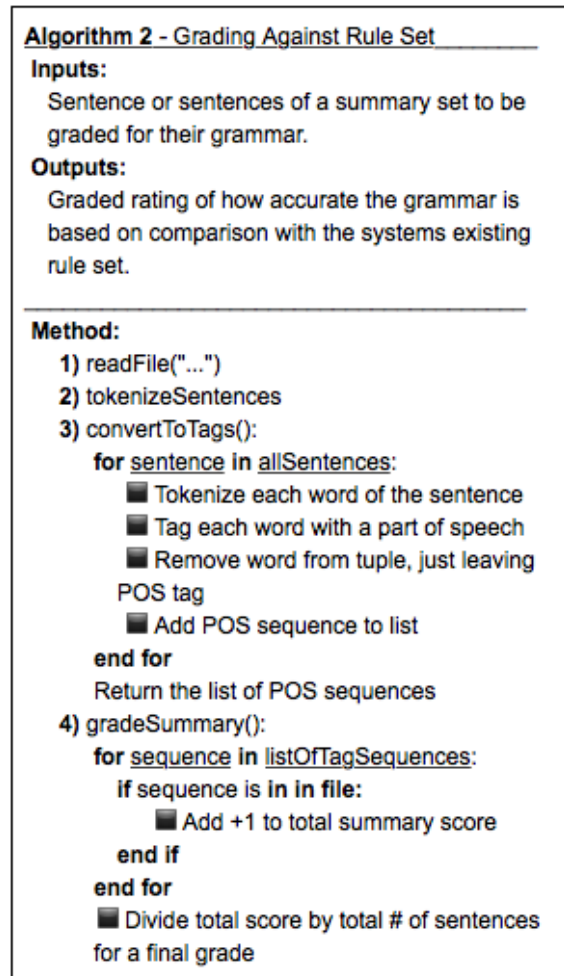

Figure 3: Algorithm 2 - Grading Against Rule Set

It will next loop through each sequence generated from the summary or set of sentences and search for a matching sequence in the established ruleset. A binary search is performed on the ruleset and if a match is found, the sentence will receive a score of '1'. If a sentence does not find a matching sequence, it is assumed grammatically incorrect and thus will receive a score of '0.' Summing up all the scores across the input sentences, it is then divided by the total number of sentences to get an average score. This final score is the given grade for the summary's grammatical correctness.

7

5 Results

We expect that through grading summaries, grammatically correct sentences will have resulted with a score of '1' and grammatically incorrect sentences scored with a '0'. As the ruleset grows with the addition of more books, we also expect to see less false-negatives.

We created three different text files to run through our grader algorithm. One is a paragraph from Jane Austen's *Pride and Prejudice* [Appendix B.1]. The second file is a file we created with sentences that have bad grammar [Appendix B.2]. The third file is a file created from sentences used in [1] as well as a few sentences we created [Appendix B.3].

The ruleset was initially established with the input of ten classical books from Project Gutenberg [Appendix A: 1 - 10] resulting in 56,025 unique POS tag sequences to be used for grammatical comparison and grading. Grading showed that it would score [Appendix B.1] correctly (Score = 1), all the sentences in [Appendix B.2] as incorrect (Score = 0), and [Appendix B.3] as only a little correct (Score = 0.166). Out of the 12 sentences in [Appendix B.3], only "I slept." and "Go!" received a score of 1. From these preliminary results, it shows that sentence structures that are known to be within the ruleset are graded appropriately. However, the ruleset was not robust enough to cover everything and thus there were some known grammatically correct sentences that were still being given a score of '0.'

The ruleset was then built upon dramatically resulting in 527,250 unique POS tag sequences. The files outlined in Appendix B were run again. Like the first round of testing, [Appendix B.1] was scored as grammatically correct (Score = 1) and all the sentences in [Appendix B.2] as incorrect (Score = 0). We achieved a slight improvement with the grading of [Appendix B.3]. The overall score increased to 0.25. In addition to "I slept." and "Go!" being graded as correct (1's), "I like carrot cake." was too.
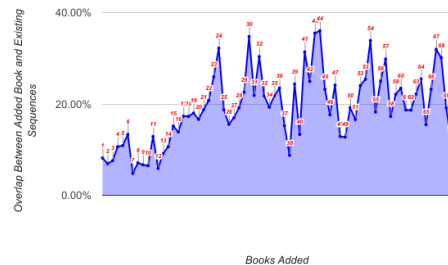
6 Limitations

Figure 4: Book Overlap - Numbers correspond to books listed in Appendix A.

When creating the text file of correct POS tag sequences, it became apparent that it would continue to grow significantly no matter how many sentences were already included in the file. While there is overlap between new books and the existing dataset, we found there was not an increasing trend in overlap (Figure 4). This could be the result of not using enough material (i.e. more

books need to be added) or because of the vast number of possible grammatically correct sentences.

While it could be that adding more sentences would yield a better result, we think this is a clear example of the infiniteness of language. When discussing sentence construction in the NLTK documentation [6], they comment on how easy it is to "extend sentences indefinitely." They go on to say "it's not hard to concoct an entirely novel sentence, one that has probably never been used before in the history of the language." Because of the infiniteness of language, grammatically correct sentences that are not in the list of grammatically correct sequences is entirely possible.

Even though we think strongly that the poor results are because of the infiniteness of language, the choice of literature used could contribute to the overall results. The only genre we used was fiction. We might see an improvement in our results if we include other genres into our dataset. Different genres have varying styles so it potentially would increase the variety of sentence structures in our dataset.

Another limitation that will start happening with this method as more sentences are added is the text file will continue to grow. As mentioned, the number of tag sequences currently in the file is 527,250 and that is only after utilizing about 70 books. As new works are added and the file grows,

there will be an increasing need to optimize the algorithms for better storage and more efficient searching and sorting.

7   Conclusion

In the NLP community there has been a lot of growth in both extractive and abstractive text summarization. One subfield of NLP that has not been matching the same growth though is being able to grade these summaries for correct grammar without the use of human graders. In this paper we put forward a methodology to automatically grade the grammaticality of these summarizations using the NLTK library. This technique revolved around growing a large, robust ruleset of known correct grammar and using this for comparison against the grammar of the summaries' sentences. The advantages of this methodology was to keep the grading automated and objective in order to replace the human factor for judging the correctness of grammar. The limitations that were run into however were not trivial. The number of possible and common grammar sequences turned out to be higher than initially expected resulting in some less than ideal results. The biggest issue being the algorithm was not able to correctly grade all correct sentences presented as correct sentences. As more books were added to the ruleset, we were able to see a slight improvement in the results but the percentage of overlap between the new book and the existing ruleset was

never consistent. Taking into account the limitations of the literature used, it is possible that incorporating more genres will increase the overlap and therefore provide an interesting scoring mechanism for the NLP community.

8   Future Work

In the future, we hope to further decrease the false negative rate by increasing the number of books used as well as incorporating different types of writing (i.e. news articles, biographies, non-fiction, etc.). It would also be beneficial to implement a different scoring system that weighs grammaticality on a scale versus scoring it completely correct or incorrect. We think that grading on a scale, meaning a sentence can be partially correct, would decrease the average number of false negatives.

It would also be interesting to do an investigation into what sentences structures overlap between the pieces of literature. We might gain clues as to what we are missing and perhaps help us create a ruleset based on patterns versus complete sentence structures. As stated previously, it is easy to concoct new sentence structures with the use of conjunctions so it could be beneficial to build a rule set on simple sentences and grade new sentences by dividing it into parts based on the location of the conjunction.

In addition, we think it will be important to experiment with additional tagging systems, mentioned in Section 3.1. Currently, we are using a very simple tagset. Our hypothesis was that the more simple the tag set, the more general sequences we would get, meaning more overlap in rule sequences so it would be interesting to test this hypothesis by using the other tagsets.

Another aspect to investigate is grammar trees. As expressed by [5], grammar trees "are integral to many computational applications, including grammar checking." NLTK has a modules that allow you to break down a sentence into its component parse tree. Instead of focusing on strictly the POS tag sequences, comparing tree structures could prove to be a better option. The individual tags could still be compared to but it would also give us the option to look at parent nodes for comparison.

In sum, more work needs to be done to grade sentences correctly but there are certainly many options to research.

References

[1] C. Li, Y. Liu, F. Liu, L. Zhao, and F. Weng, "Improving multi-documents summarization by sentence compression based on expanded constituent parse trees," *Empirical Methods in Natural Language Processing*, pp. 691–701, October 2014.

[2] J.-g. Yao, X. Wan, and J. Xiao, "Recent advances in document summarization," *Knowledge and Information Systems*, pp. 1–40, 2017.

[3] K. Woodsend and M. Lapata, "Multiple aspect summarization using integer linear programming," *Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 233–243, July 2012.

[4] R. Vadlapudi and R. Katragadda, *Quantitative Evaluation of Grammaticality of Summaries*, pp. 736–747. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[5] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Pearson Education, Inc., 2009.

[6] S. Bird, E. Klein, and E. Loper, "Natural language processing with python," 2009.

[7] T. Briscoe, J. Carroll, and R. Watson, "The second release of the rasp system," *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pp. 77–80, 2006.

[8] P. Gutenberg, Retrieved May 1, 2017, from www.gutenberg.org.

Appendices

A Literature

Listed are the pieces of literature used in the created of the tag sequences file. Literature denoted with a "G" was obtained from Project Gutenberg. Anything denoted with a "P" indicates a book from one of the author's personal collection. The literature is ordered in the way it was added to the tag sequences file.

1. Carroll, Lewis. (2008). *Alice in Wonderland.* Urbana, Illinois: Project Gutenberg. Retrieved April 1, 2017, from http://www.gutenberg.org/ebooks/11. G.

2. Bronte, Charlotte. (1998). *Jane Eyre.* Urbana, Illinois: Project Gutenberg. Retrieved April 26, 2017, from http://www.gutenberg.org/cache/epub/1260/pg1260.txt. G.

3. Twain, Mark. (2004). *The Adventures of Tom Sawyer.* Urbana, Illinois: Project Gutenberg. Retrieved April 26, 2017, from http://www.gutenberg.org/files/74/74-0.txt. G.

4. Melville, Herman. (2001). *Moby Dick.* Urbana, Illinois: Project Gutenberg. Retrieved April 1, 2017, from http://www.gutenberg.org/files/2701/2701-0.txt. G.

5. Wilde, Oscar. (1994). *The Picture of Dorian Gray.* Urbana, Illinois: Project Gutenberg. Retrieved April 1, 2017, from http://www.gutenberg.org/cache/epub/174/pg174.txt. G.

6. Dickens, Charles. (1994). *A Tale of Two Cities.* Urbana, Illinois: Project Gutenberg. Retrieved April 26, 2017, from http://www.gutenberg.org/cache/epub/174/pg174.txt. G.

7. Gilman, Charlotte Perkins. (1999). *The Yellow Wallpaper.* Urbana, Illinois: Project Gutenberg. Retrieved April 26, 2017, from http://www.gutenberg.org/cache/epub/1952/pg1952.txt. G.

8. Austen, Jane. (1994). *Emma.* Urbana, Illinois: Project Gutenberg. Retrieved April 26, 2017, from http://www.gutenberg.org/files/158/158-0.txt. G.

9. Austen, Jane. (1998). *Pride and Prejudice.* Urbana, Illinois: Project Gutenberg. Retrieved April 1, 2017, from http://www.gutenberg.org/files/1342/1342-0.txt. G.

10. Austen, Jane. (1994). *Sense and Sensibility.* Urbana, Illinois: Project Gutenberg. Retrieved April 26, 2017, from http://www.gutenberg.org/cache/epub/161/pg161.txt. G.

11. Barrie, J. M. (2008). *Peter Pan.* Urbana, Illinois: Project Gutenberg. Retrieved April 28, 2017, from http://www.gutenberg.org/files/16/16-0.txt. G.

12. Wells, H. G. (2004). *The Time Machine.* Urbana, Illinois: Project Gutenberg. Retrieved April 28, 2017, from http://www.gutenberg.org/cache/epub/35/pg35.txt. G.

13. Stevenson, Robert Louis. (2006). *Treasure Island.* Urbana, Illinois: Project Gutenberg. Retrieved April 28, 2017, from http://www.gutenberg.org/cache/epub/120/pg120.txt. G.

14. James, Henry. (1995). *The Turn of the Screw.* Urbana, Illinois: Project Gutenberg. Retrieved April 28, 2017, from http://www.gutenberg.org/files/209/209-0.txt. G.

15. Rowling, J. K. (1997). *Harry Potter and the Sorcerer's Stone.* New York, NY: Scholastic Press. P.

16. Rowling, J. K. (1999). *Harry Potter and the Chamber of Secrets.* New York, NY: Scholastic Press. P.

17. Rowling, J. K. (1999). *Harry Potter and the Prisoner of Azkaban.* New York, NY: Scholastic Press. P.

18. Rowling, J. K. (2000). *Harry Potter and the Goblet of Fire.* New York, NY: Scholastic Press. P.

19. Rowling, J. K. (2003). *Harry Potter and the Order of the Phoenix.* New York, NY: Scholastic Press. P.

20. Rowling, J. K. (2005). *Harry Potter and the Half-Blood Prince.* New York, NY: Scholastic Press. P.

21. Rowling, J. K. (2007). *Harry Potter and the Deathly Hallows.* New York, NY: Scholastic Press. P.

22. Gaiman, Neil. (2010). *Stardust.* New York, NY: HarperTeen. P

23. Bradbury, Ray. (2008). *Fahrenheit 451.* Moline, IL: Moline Public Library. P.

24. Bradbury, Ray. (2014). *Something Wicked This Way Comes.* New York, NY: Harper. P.

25. Card, Orson Scott. (2003). *Speaker for the Dead.* Princeton, NJ: Princeton. P.

26. Collins, Suzanne. (2008). *The Hunger Games.* New York, NY: Scholastic Press. P.

27. Collins, Suzanne. (2009). *Catching Fire.* New York, NY: Scholastic Press. P.

28. Collins, Suzanne. (2010). *Mockingjay.* New York, NY: Scholastic Press. P.

29. Fitzgerald, F. Scott. (1995). *The Great Gatsby.* Ware: Wordsworth. P.

30. Gaimain, Neil. (2014). *Neverwhere.* New York, NY: Harper. P.

31. Card, Orson Scott. (2009). *Ender's Game.* New York, NY: Marvel Publishers. P.

32. Gaiman, Neil. (2003). *American Gods.* New York, NY: Perennial. P.

33. Card, Orson Scott. (2002). *Ender's Shadow.* New York, NY: Starscape. P.

34. Card, Orson Scott. (2002). *Children of the Mind.* New York, NY: Starscape. P.

35. Herbert, Frank. (2005). *Dune.* New York, NY: Ace Books. P.

36. Lowry, Lois. (1993). *The Giver.* Boston, MA: Houghton Mifflin. P.

37. McEwan, Ian. (2002). *Atonement: A Novel.* New York, NY: Nan A. Talese/Doubleday. P.

38. Larson, Erik. (2003). *The Devil in the White City: Murder, Magic, and Madness at the Fair that Changed America.* Princeton, NJ: Princeton. P.

39. King, Stephen. (1977). *The Shining.* New York, NY: Anchor. P.

40. King, Stephen. (2006). *Cell.* London: Hodder & Stoughton. P.

41. King, Stephen. (2004). *Carrie.* New York, NY: Doubleday. P.

42. King, Stephen. (2001). *Bag of Bones.* New York, NY: Scribner. P.

43. Hemingway, Ernest. (2003). *The Sun Also Rises.* New York, NY: Scribner. P.

44. Hemingway, Ernest. (1995). *A Farewell to Arms.* New York, NY: Scribner. P.

45. Huxley, Aldous. (2007). *Brave New World.* London: Grafton Books. P.

46. Irving, John. (2002). *A Prayer for Owen Meany: A Novel.* New York, NY: Modern Library. P.

47. Jordan, Robert. (2000). *The Eye of the World.* New York, NY: Tor Books, LLC. P.

48. Mitchell, Margaret. (2007). *Gone with the Wind.* New York, NY: Scribner. P.

49. Kingsolver, Barbara. (2007). *The Poisonwood Bible: A Novel.* New York, NY: Harper Perennial. P.

50. Palahniuk, Chuck. (2005). *Choke.* New York, NY: Anchor Books. P.

51. Verne, Jules. (2006). *Twenty Thousand Leagues Under the Sea.* Mineola, NY: Dover Publications. P.

52. Verne, Jules. (2008). *Journey to the Center of the Earth.* Vancouver, BC: Engage Books. P.

53. Toole, John Kennedy. (2007). *A Confederacy of Dunces.* New York, NY: Grove Press. P.

54. Sanderson, Brandon. (2010). *Mistborn: The Final Empire.* New York, NY: Tor Books. P.

55. Tolkien, J. R. R. (2003). *The Lord of the Rings: The Fellowship of the Ring.* Boston, MA: Houghton Mifflin. P.

56. Steinbeck, John. (2006). *The Grapes of Wrath.* New York, NY: Penguin Group. P.

57. Salinger, J. D. (2010). *The Catcher in the Rye.* New York, NY: Back Bay Books. P.

58. Sagan, Carl. (2016). *Contact.* New York, NY: Simon & Schuster. P

59. Pullman, Philip. (2001). *The Golden Compass: His Dark Materials.* New York, NY: Knopf Books for Young Readers. P.

60. Pullman, Philip. (2001). *The Subtle Knife: His Dark Materials.* New York, NY: Knopf Books for Young Readers. P.

61. Pullman, Philip. (2001). *The Amber Spyglass: His Dark Materials.* New York, NY: Knopf Books for Young Readers. P.

62. Rand, Ayn. (2016). *Anthem.* Littleton, Colorado: Coterie Classics. P.

63. Plath, Sylvia. (2015). *The Bell Jar.* New York, NY: Harper Collins Publishing. P

64. Palahniuk, Chuck. (2006). *Fight Club: A Novel.* New York, NY: W. W. Norton & Company, Inc. P.

65. Orwell, George. (2003). *1984.* New York, NY: Houghton Mifflin Harcourt. P.

66. Vonnegut, Kurt. (2010). *Slaughterhouse-Five.* New York, NY: RosettaBooks. P.

67. Sanderson, Brandon. (2007). *The Well of Ascension.* New York, NY: Tom Doherty Associates, LLC. P.

68. Sanderson, Brandon. (2008). *The Hero of Ages*. New York, NY: Tom Doherty Associates, LLC. P.

69. Abagnale, Frank W. (1980). *Catch Me If You Can*. New York, NY: Grosset & Dunlap. P.

70. Alcott, Louisa May. (2004). *Little Women*. New York, NY: Barnes & Noble Books. P.

B Graded Sentences

Sentences used for grading.

B.1 Excerpt from Book in Ruleset

From Jane Austen's *Pride and Prejudice*.

The evening altogether passed off pleasantly to the whole family. Mrs. Bennet had seen her eldest daughter much admired by the Netherfield party. Mr. Bingley had danced with her twice, and she had been distinguished by his sisters. Jane was as much gratified by this as her mother could be, though in a quieter way. Elizabeth felt Jane's pleasure. Mary had heard herself mentioned to Miss Bingley as the most accomplished girl in the neighbourhood; and Catherine and Lydia had been fortunate enough never to be without partners, which was all that they had yet learnt to care for at a ball. They returned, therefore, in good spirits to Longbourn, the village where they lived, and of which they were the principal inhabitants. They found Mr. Bennet still up. With a book he was regardless of time; and on the present occasion he had a good deal of curiosity as to the event of an evening which had raised such splendid expectations. He had rather hoped that his wife's views on the stranger would be disappointed; but he soon found out that he had a different story to hear.

B.2 Poor Grammar

Sentences made by the authors with bad grammar. The third sentence is from [Appendix A: 11] with the sentence jumbled. The fifth sentence is from [Appendix A: 11] with improper verb agreement.

The teapot car blue spits cloud. The moon cow jumped over the. All except one children, grow up. He walk to the store every Monday and Thursday. The rabbit-hole went straight on

like a tunnel for some way, and then dips suddenly down, so suddenly that Alice had not a moment to think about stopping herself before she found herself falling down a very deep well.

B.3   Sentences Used in Original Paper

These sentences were mentioned in [1] when using their approach at summarizing sentences. There are a couple of source sentences, human compressed sentences, and also algorithm compressed sentences. In addition, we created the last three sentences to see how simple sentences would score.

Apart from drugs, detectives believe money is laundered from a variety of black market deals involving arms and high technology. Detectives believe money is laundered from a variety of black market deals. Apart from drugs detectives believe money is laundered from a black market deals involving arms and technology. Detectives believe money is laundered from a variety of black deals involving arms. Detectives believe money is laundered from black market deals. Mrs. Allan's son disappeared in May 1989, after a party during his back packing trip across North America. Mrs. Allan's son disappeared in 1989, after a party during his trip across North America. Mrs. Allan's son disappeared in May, 1989, after a party during his packing trip across North America. Mrs. Allan's son disappeared in May 1989, after a party during his trip. I slept. I like carrot cake. Go!