Simply5: A Proposal for a Text Simplification System for Effective Learning

Akshay Dangare
Department of Computer Engineering
Vidyalankar Institute Of Technology
Wadala, Mumbai
dangareakshay@gmail.com

Meghna Pai
Department of Computer Engineering
Vidyalankar Institute Of Technology
Wadala, Mumbai
meghnapai0312@gmail.com

Rohit Kothavade
Department of Computer Engineering
Vidyalankar Institute Of Technology
Wadala, Mumbai
rohit.kothavade@gmail.com

Mohini Kamat
Department of Computer Engineering
Vidyalankar Institute Of Technology
Wadala, Mumbai
mohini.kamat@vit.edu.in

***Abstract*—We live in an era of information overload, wherein unimaginable amounts of text are available to us in different forms at the click of a button. However, it is often experienced that this information is not always available in the simplest of forms and also comes embedded with many jargons. The manner in which a text is written—its vocabulary, its structure—can be difficult to read and understand for many people, especially those with poor literacy, cognitive or linguistic impairment, or those with limited knowledge of the language of the text. Hence, the vocabulary and the syntax of the text are the two barriers we come across that make reading text really difficult. This paper proposes a tool called 'Simply5', which is a Text Simplification system that tries to tackle the problem of understanding complex text and simplify Jargons that make reading difficult. The process of Text Simplification is nothing but the transformation of one piece of text into another, ideally conveying the same meaning, but in a much more simplified manner and which is easy for a layman to understand. Although some research has been done on Text summarization in the growing field of Natural Language Processing, there is no such universal system available for the user wherein he can simplify the text he cannot understand. Through Simply5, we wish to provide the user a handy tool that can be used to simplify any chunk of text, anytime, anywhere on the go.**

Keywords—Text implification Tool; Lexical Substitution; Natural Language Processing; Sentence Generation; Syntactic Simplification; Word Sense Disambiguation.

I. INTRODUCTION

Thanks to the availability of texts on the Web in recent years, increased knowledge and information have been made available to broader audiences [22]. People now-a-days, look for answers to the most trivial questions, online. Naturally, the question arises that how much of the stuff you read online, do you actually understand? This is due to the variation and unpredictability of the English language. Firstly, different people have different habits of writing, leading to different structure of text. For example, some like to split up their thoughts into multiple simple sentences, whereas some people like to write long and complex sentences to produce a better effect. Similarly, the same word may have a different meaning when used in a different context. Therefore, the way in which a text is written—its vocabulary, its syntax—can be difficult to read and understand for many people including children, language learners, the elderly, the hearing impaired and people with aphasia or cognitive disabilities or those with limited knowledge of the language of the text, as explained in [1, 22]. Texts containing uncommon words also called as jargons or some long and complicated sentences can be difficult to comprehend and understand by people as well as difficult to analyze by machines. As a result, the complex structure and difficult jargons often prove to be the hindrances in understanding the given text.

One solution to solve this problem is provided by the process of Automatic Text Simplification, a sub-topic in the domain of Natural Language Processing. Text Simplification is the process of enhancing, modifying and simplifying the grammar and the structure of the given text, such that the underlying meaning and the logical structure remains the same. Some of the ways in which this can be done is by replacing the difficult or unknown phrases with simpler equivalents or by transforming the long and syntactically complex sentences into shorter and less complex ones. Text Simplification, a research topic which started two decades ago, now has taken on a pivotal role in natural language processing not only because of the interesting challenges it possesses but also because of its social implications, and the huge problems it can tackle owning to the exponential increase in the amount of data produced. [22]

However, although extensive research has been conducted on simplifying small quantities of text, and using certain approaches that help solve only specific type of inputs, using a particular algorithm. A summary of the various text simplification techniques are thoroughly presented in [2]-[4]. Extensive work on simplifying the structure of sentences has been done in [5, 6] and both produce satisfying results in reducing the complexity of the long and difficult sentences. However, no effort has been made to simplify the jargons and replace them with simplified set of words in this one. Similarly, [4, 8] deals with simplifying the difficult words in a sentence, without modifying the syntax of the complex sentences thus making them easier to understand. Even if both the problems are addressed together [4], no such application is available to the user that can input text and produce an equivalent output text where the vocabulary and sentence structure are simpler.
[23] is the only system available on the web, open to the user which does text simplification. However, this too only replaces the difficult vocabulary with their meanings, without improving the sentence structure, nor converting the meaning that fits in the context.

Through this paper, we wish to combine the various text simplification techniques in order to address the above mentioned problem by introducing **Simply5 - a naval Text Simplification tool** that will allow the user to input a block of text in English, and will try to simplify the same both in terms of its structure - by breaking down complex sentences into smaller and simpler ones, and its vocabulary - by identifying difficult words as per user's level expertise, and then tries to replace them with simpler set of words that easy to comprehend.

II. Problem Statement

[2] Adult literacy is a concern in developed and developing countries; for instance, one in six adults in the UK have poor literacy skills 1, and only a quarter of Brazilians who have studied for 8 years can be considered fully literate [20]. In this process education, whenever a learner of any category or age group tries to learn English or tries to understand the given sentences or comprehensions, the process of understanding becomes difficult when the sentences consists of complex and high-level words. Considering today’s news articles, scientific journals, the level of vocabulary used is quite complex and each often one comes across some words which he/she cannot understand without using the dictionary. We believe that Simply5 would provide an innovative solution to tackle this problem as the user can just enter the text that he cannot understand i.e. a paragraph or a sentence into the input box and get a simplified version of the input at a click of a button.

The motivation behind coming up 'Simply5' was to not only develop a multi-layered text simplification system as described below, but more importantly to present an accurate, robust and a reliable tool to the world, the first of its kind. Through the use of Simply5, we wish to promote the incorporation our software in user's daily tasks such as reading or writing a mail, reading an article or a newsletter, reading a blog, or an e-book. Whenever the user experiences any difficulty while going through any of these, he should be able to immediately copy the difficult part, into our system and expect a simplified output.

Step 1: Identifying a difficult word.

Step 2: Searching for its meaning in the dictionary.

Step 3: Determining the correct meaning depending upon the context in which the word was encountered.

Step 4: Replacing the word in the parent text with its meaning in such a way that the overall semantics make sense.

We are proposing a three-layered text simplification system that incorporates multiple text simplification techniques to ensure the simplified text delivered to the user is of the purest form in terms of vocabulary as well as syntax. The characteristics of this tool, are exemplified by realizing the applications where it can be employed and can serve the different type of users.

III. METHODOLOGY

The proposed Text Simplification system would be based on a "Three-layered Architecture" as explained below. Once the user inputs the text he wishes to simplify, it will first go through the 3 layers sequentially to finally come out in a simplified form. Syntactic Simplification [5, 6], Lexical Simplification [8] and Sentence Generation [10]-[16] are the three Text Simplification techniques we will be using in Simply5 in coordination to ensure that the text thus obtained is in the purest form free from jargons and syntactic complexities, and makes senses semantically.

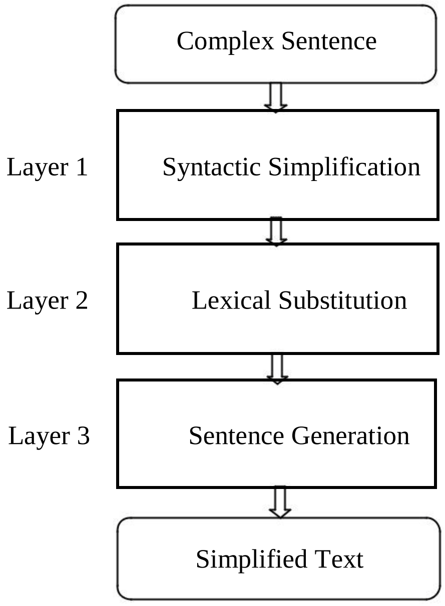


Fig. 1. Proposed Three-Layered Architecture for Simply5A

A. *Input Complex Text*

The first stage in the process of Text Simplification is to allow the user to place the text he wants to simplify, into an input text box. This can be done by merely copy-pasting the complex text into the textbox, on the go whenever the user comes across one, while reading any material. Now the input can be either a single sentence or a large chunk text having multiple sentences. In this stage, once the user's input is captured, our system would split up the entire block of text, into separate sentences and store them individually in a suitable data structure. This dataset can be considered as the input repository from which, each sentence would be sent sequentially to the following three layers to undergo the process of Simplification.

B. *Syntactic Substitution*

The first layer of simplification in the Three-layer Architecture is Syntactic simplification. This process induces significant improvements in structure of the text. Syntactic Simplification can be defined as the process that makes a block of text easier to understand by humans or programs by reducing the syntactic complexity of a text at the same time preserving its meaning and information content [6]. Commas, coordinating conjunctions, and adjacent comma–conjunction pairs are considered to be potential indicators of coordination in natural language. In the below example explaining Syntactic Simplification, sentences containing classified potential coordinators are converted into sequences of simple sentences by breaking them down.

Example:

Jones, who was the CP of a club, entered the hall.

Syntactically Simplified text:

Jones entered the hall. Jones was the CP of a club.

Simplification such as above which takes in the syntactically complex text and processing it to give simplified one which is semantically equivalent is done in 3 phases as shown-

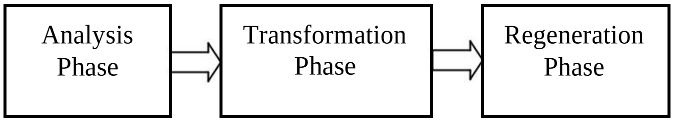


Fig. 2. Syntactic Simplification process Pipeline [4]

i) Analysis Phase

The analysis phase is the initial phase and is multifunctional. It breaks down the text into sentences. This is the pre-requisite to the next phase because the syntactic-simplification rules work at the sentence level. It further marks-up sentence structures like elementary noun phrases that can be simplified in each sentence. Words are parts of speech (POS) tagged. It also includes a pronoun-resolution

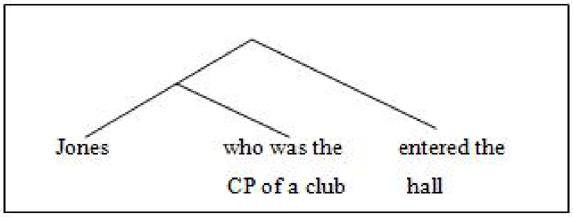


Fig. 3.Analysis Phase example

ii) Transformation Phase

The main function of the transformation phase is to simplify the analyzed text syntactically. Here, we use our own customized rules. The transformation stage uses recursion to simplify the analyzed text and this text is represented using a stack with the first sentence on the top. This stack then undergoes a recursive transformation as shown below.

Popping the first sentence from the stack is the initial step at each transformation stage. When there are no simplifiable constructs left i.e. when the base case is reached, it is appended to the end of an output queue. In recursive case, when a popped sentence is decomposed into two simple sentences using the transformation rule, those simplified sentences are sent to the regeneration phase where the issues of conjunctive cohesion are solved. These two regenerated sentences are then again pushed on the top of the transformation stack in the specific order given by the regeneration stage. A top-down simplification approach is used when a single sentence consists of multiple constructs that can be simplified. Once the transformation stack is empty, the simplified text is entailed to the output queue. The transformation stage invokes the regeneration stage for fixing the pronominal links on the output queue and then the simplified text is produced as an output. [6]

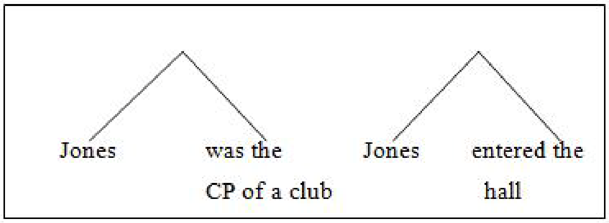
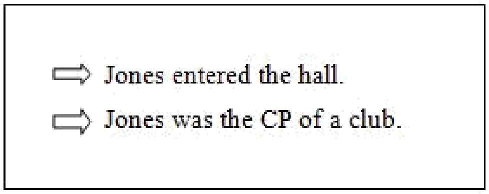


Fig.4.Transformation Phase example.

iii) Generation Phase

The motivation of syntactic simplification in making a text accessible to a broader audience can be undermined if the regenerated text lacks cohesion. And hence the generation phase is very crucial. This phase does the functions like deciding sentence order, introducing cue words, preserving anaphoric links. The syntactically simplified sentences for the given example are formed as shown below in Fig 5.



⇨ Jones entered the hall.

⇨ Jones was the CP of a club.

Fig. 5.Generation Phase example.

C. *Lexical Substitution*

Whenever a learner of any category or age group tries to learn English or tries to understand the given sentences or comprehensions, the process of understanding becomes difficult when the sentences consists of complex and high-level words. To understand the meaning the user refers dictionary and gets multiple synonyms for the same. However, not every synonym can fit in the sentence to give the original meaning. The user then needs to understand the context in which those synonyms can be used and then find the most appropriate synonym to replace the complex word in the sentence. This is the case when the user comes across just one such complex word.

If we think of today's news articles, scientific journals, competitive exam preparation material, the level of vocabulary used is quite complex and each and every individual comes across some words which he/she cannot understand without using the dictionary. To avoid the 4-step problem as explained in the problem statement above, our system would provide a much better alternative wherein the user can just enter the entire piece of text into the input box and get a simplified version of the input at a click of a button.

The part wherein the high-level word would be replaced with their appropriate simpler synonyms is called as Lexical Substitution. Word Sense Disambiguation plays an important role when the process of selecting an appropriate synonym from the list of all the synonyms is to be done.

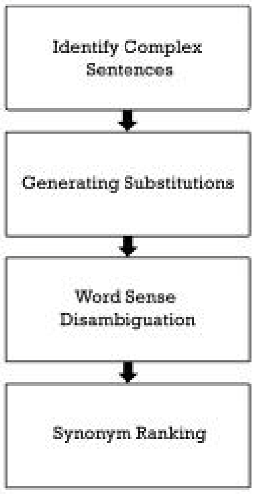


Fig. 6. Lexical Simplification Pipeline that will be employed in Simply5, along an example to explain the working of the process. [4]

Example – While a large number of Hong Kong companies have reincorporated offshore ahead of 1997, such a move isn't an option for Cathay because it would jeopardize its landing rights in Hong Kong.

i) **Identifying Difficult words –** Initially, the system would find all the difficult words in the given text which the user might have a problem understanding. In the above example,

Complex words – reincorporation, jeopardize

ii) **Generating Substitutions –** Now that the complex words are identified, next it would look up the dictionary available in the database, to generate a set of possible substitutions for the given word.

Reincorporation = back to an original place or condition, set up again, get back together.

Jeopardize = endanger, cause harm, put in danger.

iii) **Word Sense Disambiguation –** It is a process wherein the synonyms of the complex words which does not fit the context are eliminated and then a ranking algorithm is applied.[9] There are several algorithms for both the processes. Coming to the given example, the following set of meanings would be discarded by the system.

Reincorporation = back to an original place or condition, ~~set up again~~, get back together.

Jeopardize = endanger, ~~cause harm~~, put in danger.

iv) **Synonym Ranking –** As mentioned above, once the meanings which do not fit the context are removed by Word Sense Disambiguation, next step is to rank the available set of meanings as according to the context, so as to select the best possible meaning from the set. This can be done by the various Synonym Ranking Algorithms as explained in [9].

Reincorporation-
#1.Move back to an original place or condition #2.Get back together.

Jeopardize-
#1.Endanger
#2.Put in danger.

v) **Generating Simplified Text** – Finally, once the synonym ranking algorithm gives us the best meaning from the Synset, we can now substitute it in the input to generate a simplified output.

Output-

While a large number of Hong Kong companies have **move back to an original place** at offshore ahead of 1997, such a move isn't an option for Cathay because it would **endanger** its landing rights in Hong Kong.

Finally, this above output is what is produced at the end of the Lexical Simplification phase. However, as you might have noticed that after substituting the meaning of ‘reincorporation’ in the parent text in its place, the overall sentence does not grammatically correct. “While large number of Hong Kong companies have **move back to an original place**”. But this isn’t the case with the other word that was substituted, as it was a single word substitution of ‘jeopardize’ with ‘endanger’ which fit the context perfectly. Hence, we can conclude that even though Lexical Simplification greatly helps in simplifying the complex words, it does not always produce grammatically correct sentences. This is finally rectified by the process of ‘Sentence Generation’.

D. Sentence Generation

[10] Sentence generation, also called as Natural Language Generation (NLG) in the field of Natural Language Processing, is the computational process of automatically producing sentences in natural language based on the type of input it receives. A sentence generation component receives information about what is to be communicated as an input. This input can be of different forms depending upon the approach followed to design the system. The sentence generation component then processes the input and produces the corresponding natural language expression as an output. Recent approaches for sentence generation use logical forms, set of facts in the form of knowledge representation or a set of closely related words as in input. Thus, a generation component might accept as input like a set of words such as-

(green), (shelf), (on), (cup)

and convert this into something like the expression "The green cup is on the shelf." In natural language. The end result of such a regeneration process is quite clear: a sentence must be produced which is grammatically correct and which has made appropriate choices of words also called as lexemes. [10]

Coming to the output produced in the example mentioned in the Lexical Simplification phase, the substitution performed for ‘reincorporation’ wasn’t grammatically correct. Hence, we need to use the sentence generation process to convert-

“have **move back to an original place** at offshore”

↓

“have **moved back to an original place** at offshore”

As correctly identified, the problem in the first sentence was the tense of the verb “move”, as it was directly replaced from the meaning and needed modification. Sentence generation thus identifies such words which require subtle modifications into order to fit into the existing sentence, for the sentence as a whole to be correct grammatically.

Some early research on sentence generation indicated that the basic components of a sentence generator are similar to those in analysis or syntactic parsing that comprise of a grammar and a lexicon [5]. The earliest sentence generators to be built as shown in [15, 16] used only these minimal components. This restricted them to the random generation of grammatically correct sentences since specifications of intended communicative purpose were not present.

Since then, however, we have come a long way in working on Natural Language Generation (NLG) systems that produce sentences of desirable quality. [10] comprehensively explains the various sentence generation techniques and algorithms, and gives quantitative comparisons for each. [11] introduces a NLG system called as “VINCI” which accepts as input a formal description of some subset of a natural language, and generates strings in the language, using Linguistic Modelling technique. Advances in Artificial Intelligence have enabled the extensive use of Neutral Networks in NLG systems, as explained in [12], which makes use of a neural net approach to generation.

Thus, owing to the various techniques available to generate simplified text, to further improve sentence construction both semantically, as well as grammatically, we can safely predict that the simplification thus produced by going through the Three-Layered Architecture in Simply5 would have a much improved accuracy and simplified structure and meaning.

E. Simplified Text

Finally, after passing all the sentences stored in the input repository created in the first stage (Section A) through the three stages mentioned above, we can say that the input text has been completely simplified by our system. This simplified text can be tested against various parameters such as accuracy, simplification of sentences with complex structure, replacing the difficult words with their simpler meanings as well as fitting the meanings into the context to provide output text. This output text thus produced is expected to be correct grammatically as well as semantically. This will be the methodology that will be implemented in Simply5, to help us understand the language better.

IV. The final product

We envision Simply5 to be a desktop application, installed and available on the user's system offline. This allows the user to use Simply5 simultaneously while reading an article, mail, story etc. without the need of being connected to the internet, as in the case of website. [24] gives a brief summary of the advantages and disadvantages of desktop applications over web based applications. Attempts would be first made to develop a stable desktop system for Windows, and later it would be ported to other platforms such as LINUX and iOS. Since majority of people own a windows desktop, this will help us reach a broader audience as compared to the others systems.

V. Applications

The main goal of Simply5 is to make information more accessible to the large number of people with reduced literacy. Apart from our system being used universally by anyone who comes across a problem understanding English, we believe that our system would especially benefit the three important sections of the society, namely the young teenagers recently introduced to English, the second language learners trying learn English and finally the ones preparing for top competitive exams such as GRE/TOEFL/GMAT and many more. Here is how the above mentioned types of people can use our system to aid their process of learning English

A. Level 1 : Children and Young Students

Children, especially those who have just earn the basics of the English language, in class 5 to 8, could use Simply5 to practice their newly learnt English skills that they learnt in school. This can be done making them read a simple and short stories in English, try to understand it, and then reading the simplified version of the same to realize the parts which they were unaware of. One important motivation while coming up Simply5 was to come up with a solution that could assist such young minds, by speeding up their learning process at an early stage of their education.

B. Level 2 : Second Language Learners

Although English is considered as a global language with a majority of countries speaking it, there are still may people for whom English is not their primary language. Such people, who are trying to learn a new language apart from their mother tongue, are called as Second Language Learners. These are people who have learnt the language at an intermediate level, and now want to master it by speaking fluently and understanding correctly. To do this, such people start reading books, articles, online news for improving their reading and understanding. This is exactly where Simply5 can help them by expediting their process of mastering English by helping them overcome hurdles they come across in this process.

C. Level 3 : Competetive test takers

Speaking of Second Language Learners, almost all such people are required to give International Aptitude Exams such as the GRE/TOEFL/GMAT to enter into any English speaking country such as the United States. The level of vocabulary as well as sentence structure is extremely difficult in such exams, as the time allotted is extremely less. Naturally, most of the test takers struggle early on while preparation, in filling out the blank or understanding a passage. Such prospective test takers can use Simply5 to instantaneously feed the questions into it, and get a simplified version of the question, which in will instruct them the way in which the given question or passage should have been interpreted, thus providing them a way to tackle such questions in future.

VI. Conclusion

As explained through the various sections of this proposal, we need a full-proof universal Text Simplification system to overcome our linguistic barriers. Such a system must fulfill two very important objectives-

- The simplified output produced by the system must be have a simplified structure, simpler vocabulary and must be grammatically correct.
- The system should adapt to the different types of inputs given by the user, and must always be available to user without utilizing much of its resources.

The system proposed by us, called 'Simply5' stands true on both the above objectives, and would definitely benefit a large section of the society helping them learning the English language effectively and at a faster rate.

References

[1] Carroll, John, et al. "Practical simplification of English newspaper text to assist aphasic readers." Proceedings of the AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology. 1998.

[2] Siddharthan, Advaith. "A survey of research on text simplification." ITL-International Journal of Applied Linguistics 165.2 (2014): 259-298.

[3] Feng, Lijun. "Text simplification: A survey." The City University of New York, Tech. Rep (2008).

[4] Shardlow, Matthew. "A survey of automated text simplification." International Journal of Advanced Computer Science and Applications 4.1 (2014): 58-70.

[5] Chandrasekar, Raman, and Bangalore Srinivas. "Automatic induction of rules for text simplification." Knowledge-Based Systems 10.3 (1997): 183-190.

[6] Siddharthan, Advaith. "Syntactic simplification and text cohesion." Research on Language & Computation 4.1 (2006): 77-109.

[7] Kauchak, David, et al. "Text simplification tools: using machine learning to discover features that identify difficult text." System Sciences (HICSS), 2014 47th Hawaii International Conference on. IEEE, 2014.

[8] Biran, Or, Samuel Brody, and Noémie Elhadad. "Putting it simply: a context-aware approach to lexical simplification." *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics, 2011.

[9] Turney, Peter D. "A uniform approach to analogies, synonyms, antonyms, and associations." Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1. Association for Computational Linguistics, 2008.

[10] Bateman, John A. "Sentence generation and systemic grammar: an

[11] introduction." Iwanami Lecture Series: Language Sciences 8 (1997): 1-45.

[12] Levison, Michael, and Gregory Lessard. "A system for natural language sentence generation." Computers and the Humanities 26.1 (1992): 43-58..

[13] Hammervold, Kathrine. "Sentence generation and neural networks." Proceedings of the first international conference on Natural language generation-Volume 14. Association for Computational Linguistics, 2000.

[14] Shieber, S. M. (1993), 'The problem of logical-form equivalence',

[15] Computational Linguistics 19(1), 179– 190.

[16] McDonald, D. D. (1993), 'Issues in the choice of a source for natural language generation', Computational Linguistics 19(1), 191 – 197.

[17] Yngve, V. H. A. (1962), Random generation of English sentences, in 'The 1961 Conference on Machine Translation of Languages and

[18] Applied Language Analysis', Her Majesty's Stationery Office,London.

[19] Friedman, J. (1969), ‘Directed random generation of sentences’,

[20] Communications of the Association for Computing Machinery 12(6)

[21] Tanner, Ryan. "Creating a Semantic Graph from Wikipedia." (2012).

[22] Zhang, Yaoyuan, et al. "A Constrained Sequence-to-Sequence Neural Model for Sentence Simplification." arXiv preprint arXiv:1704.02312 (2017).

[23] Chaudhari, Mohini, and Sharvari Govilkar. "Emotion Extraction using Rule based and SVM-KNN Algorithm." International Journal of Computer Applications 125.11 (2015).

[24] Alu´ısio, S. M., L. Specia, T. A. Pardo, E. G. Maziero, & R. P. Fortes. 2008. Towards brazilian portuguese automatic text simplification systems.

[25] Niklaus, Christina, et al. "A Sentence Simplification System for Improving Relation Extraction." *arXiv preprint arXiv:1703.09013* (2017).

[26] http://www.morganclaypool.com/doi/abs/10.2200/S00700ED1V01Y201

[27] 602HLT032

[28] https://www.simplish.org/

[29] https://blog.computerlagoon.com/2015/08/04/pros-and-cons-to-web-and-desktop-applicat