Estimation of Lexical Complexity using Language Semantics

Ketakee Nimavat
UG student,
Computer Engineering Department,
L. D. College of Engineering, India

Tushar Champaneria
Assistant Professor,
Computer Engineering Department,
L. D. College of Engineering, India

ABSTRACT
Word complexity is a quite complex and subjective issue. However, it is also intuitive. Here the topic is explored and an intuitive method is proposed to judge the complexity where the intuition is based on the genesis and development of a language. The proposed technique is analogous to a tree structure where in each word is made up of its child nodes where child nodes signify simpler words. The algorithm hence takes into account the definition of the word and finds the complexity score based on the basic words present in the definition. The method is then judged using Flesch Reading ease and tested on separate sets of simple and difficult words. It is observed that this helps judge the complexity of a text as whole and works fairly well for individual words as well.

General Terms
Natural language Processing, Text complexity, semantic complexity, lexical complexity, Text simplification, Text summarization.

Keywords
Text Simplification, semantic complexity, lexical complexity, text complexity, lexicon.

1. INTRODUCTION
We are living in an era of ubiquitous intelligence where the intelligence knows what we want when and where. Our devices and platforms that we interact with know what books we would like to read next, which show to watch and even what we would like to say in a reply. It seems like we can design intelligence to do everything, from complete our sentences to take us places. Yet, they don't necessarily understand our world. Current techniques can process and predict very well, perhaps even better than humans[1] but when it comes to task that require understanding a system and working on the components, there is a lot that can be done. Processing information and predicting work very well even only observations of the system's behavior are available. These observations can help one get an insight into the components of the system and the relation between the components and use them to predict future behavior. The methods work well for systems that have statistical environment such as e-commerce websites or social media platforms where the knowledge about the system is quantifiable and data discreet. For now, the current methods do quite well in areas of machine translation, conversational agents, automated cars, mining information etc.[2] but for further depth and a realistic understanding in areas like language and visualization, the statistical methods perhaps might not be enough, and even if they are, they might not have an understanding of inherent structure of the system which could prove to be an obstacle in tasks of generation such as image description, summarization, simplification, empathetic decision making[3]. These areas are hard to conquer for they don't necessarily have one mathematical model to fit the patterns and simple rule based models are either too vast to build or too rigid for the real world. Especially in the field of natural language processing where in upcoming applications such as summarization, simplification, text generation, response generation all require an understanding of the system. A literature is as useful as much as it is accessible. A gold plate with the answer to answer life written in obscure symbols is probably not useful at all. Its accessibility depends upon the capability of its readers. An article with high very high language would serve no purpose for laymen readers. On the other hand, a beginner of a language might want to access high level content but can't because the language is too obscure. Simplification helps in these scenarios. Be it only for the sake of quick reading or to make a document more accessible.

A text can be complex in two ways: it could either be complex because of the inherent structure used by the writer or because of the words used. Generally, dealing with complex sentence structures could use a grammatical approach or a set of structure transition rules or even machine learning. Various such approaches can be seen in [4][5][6]. The second approach, lexical approach is the one we discuss here. Complexity induced by difficult words can be simplified by using simpler words. Simplification becomes difficult because words can be used in multiple senses and it is difficult to get a lot of coherent data on relative simplicity of words. To overcome this, an intuitive method is proposed to judge relative word complexity that doesn't require training or several example use cases but would definitely work better on integration with modern machine learning methods. The proposed method is targeted to judge the complexity of the word to aid in lexical substitution. It can however be used to judge complexities of various texts and compare them as well. The robustness of the method enables the use of complexity score in generation of simpler texts based on the user's level of competence. The coming section, section II, explains the types of complexities, the complexity the method would solve and the intuition behind our approach. In section III, the algorithm is proposed and the metrics used to judge the method are explained. Section IV describes the observations of the test and in Section V methods of usage and possible improvements are discussed. Finally, section VI concludes the paper with final words about the method and other possible ways to go from here.

2. COMPLEXITIES: LEXICAL, STRUCTURAL AND SEMANTIC
Text complexity sounds too difficult to put into metrics since language generates from a person and has no fixed structure apart from a few grammatical rules. However, we can classify it into two major kinds: ***Lexical, Structural.***

2.1 Lexical Complexity

It is the complexity that is concerned with the lexicon and is caused by the vocabulary used in the text. For example the sentence: *"Such an abominable crime"*. The meaning of the sentence is blurred because of the word "abominable" which might not be a familiar word for a lot of readers. Replacing the 'abominable' with '*terrible'* would help simplify the sentence and still maintain the meaning at large. Various ways in which lexical complexity is a hurdle are: *Usage of long words, rich words that are used for formal or academic purposes, words that are not colloquial, words that one knows only after a certain level of education i.e. words with high age of acquisition and so on.*[7][8]

We focus here on proposing a measure for determining the complexity of words that will take into factor: high age of acquisition, rich words and formal words for which local words are available.

2.2 Structural (Syntactic) complexity

The other type of complexity is structural complexity which is caused by the writer's writing style and the sentence formation. Examples of structural complexity can be seen in Classic novels and in the works of authors such as Shakespeare. It often is solved by humans and for very rich texts and requires someone with an expertise. For a majority of the people, it is too complex to understand at in one go. That being said, multiple algorithms have been proposed based on manipulating the grammar such as handcrafting grammars[5], a two staged system that involves syntactic and lexical simplification [9], narrow domain systems that work on texts of one field [10] and various modern systems do well on structural complexity after being thoroughly trained[11]. Apart from that Rule based methods can also be useful but they fail to be as robust as their human counterparts.

2.3 Semantic complexity

Here, the issue of lexical complexity is addressed, the words that make the text difficult to understand. Basic words are words that are simple and used in colloquial scenarios and are understood by majority of the readers. Complex words on the other hand are the ones that hamper the understanding of the text. A simple small word might hamper the understanding if its usage is obscure. Similarly, longer words are unlikely in everyday spoken or written dialogues hence they might be unfamiliar to laymen. On the other hand, a long word might be familiar because of its frequency of usage in everyday life. Complexity hence is considered to made up of two major components: ***Word length(How long it is)***[12] ***and Word usage (How obscure it is)***

A third measure of semantic complexity is proposed based on the following intuition:

Words are formed as the need for a word arises, with generations, the structure accumulates length as more and more words are entwined together to form the new word. Sometimes, words are borrowed from other languages as well. Complexity in text is often used to judge its readability. While it is not possible to judge the complexity of words borrowed from other languages, it is indeed possible to do so for the words generated within a language. Languages that formed as a result of two communities interacting are called Pidgins, as generations go by, the new mix language first becomes the language of a newer generation and then with each generation becomes more developed. This language that is now the native language of the newer generations is called creole. Where in, it is not fully developed like a modern language would be but neither is it brand new. It still has elements from all of its source languages and has structured grammatical patterns. [13] This pattern of language development suggests a very intuitive approach to complexity of words. A word will be as complex as the thing it is trying to express. As more interactions among basic elements are included in the narrative, the complexity of the word increases. Take the example of colors, there are certain colors for which words are always found in any language, these colors are shades of red and green and even blue. Other colors and their names emerge much later as the society grows and the language becomes more complex. We use this intuition of basic words to propose another method for determining relative complexity between two words.

3. PROPOSED ALGORITHM

As discussed in above section, a word is as complex as the number of basic words it is made of. Transferring that to the real world, it translated to: a word is as complex as the number of simple words in its definition. More importantly, word complexity increases as the number of basic words in it increase. Hence, creating a word tree like structure where a word is made up of its children, the higher the height and number of children of the tree, more complex is the word. The algorithm used is as follows:

Algorithm: Semantic Word complexity
Input: word
Output: complexity score depicting the complexity of the given word

```
1:  cp ←0
2:  defn←get_definition(word)
3:  tokens←word_tokenize(defn)
4:  useful_words←remove_stopwords(tokens)
5:  for elem in useful_words:
        if elem in basic_word_list:
                cp ← cp + 1
        else:
                defn2←get_definition(elem)
                tokens2←word_tokenize(defn2)
                useful_words ←
        remove_stopwords(tokens)
                cp← cp + len(useful_words)
6: return cp
```

Algorithm's computational complexity: The above algorithm has complexity for each of the above mentioned steps as follows:

1. $O(n)$
2. $O(1)$
3. $O(w)$ *#cost of tokenizing a sentence*
4. $O(s)$ *#cost of removing stopwords*
5. $O(n * b)$ *or* $O(n * (1 + w + s + 1))$ *#b=cost of checking against every basic word*

Hence overall complexity can be given by Computational Complexity:

$$\boldsymbol{O(w + s + nb)\ Or\ O((w + s + 2)(n + 1))}$$

(where w = cost of tokenizing the definition,
s = cost of removing stopwords,
n = length of the definition
b = cost of checking the current word against each word in the basic word list)

In the following paragraphs, various factors that affect the accuracy of the algorithm are discussed.

The algorithm proposed consists majorly of two parts: ***the basic word list, the dictionary.*** *The basic word list* consists of the base words upon which the other words are considered to be built and *the dictionary* provides the definition for the words. These two parts determine the accuracy of the algorithm and hence are very important, especially the basic word list, which determines the base level for the complexity algorithm. In the testing of this algorithm, basic words are the words that appear in OgDen's 2000 word list and 1000 basic words from Wikipedia.[14][15]

Stopwords: are the words that are often used in English and can be removed without any harm to the meaning of the sentence. Words such as 'the', 'a',' an' etc. fall under this category. The kind of list used will determine the accuracy and the correctness of the algorithm.[16] The one used here is from Python nltk library.[17]

get_definition(): gets the definition of the word from the dictionary. It returns a string which consists of the definition.

word_tokenize(): splits the definition into its consisting words and returns a list of words in the definition.

remove_stopword(): returns a list of words from the definition word list that are not stopwords.

len(): returns the length of the element given to it. Here it returns the number of items in the list.

cp: is the complexity score. Higher the score, more semantically complex is the element.

The depth of the algorithm in terms of iterations is limited to two and in the second iteration, instead of counting the number of basic words in the definition, the depth of two is considered instead of number of basic words in every definition of every non-basic word since counting the number of basic words recursively in every definition is prone to infinite looping.

The length is used instead of the number of basic words in the last layer because the definition of the word might be made up of more complex words than basic words in which case, using the basic words for the second level might give inaccurate results. Using the length of the definition, meaning the number of words in the definition, hence makes up for the flaw. It has proved to be a better measure of complexity in testing.

Finally, to determine the complexity of full texts, the complexity score of each word is added and it is divided by the number of words in the document other than the stopwords.

How is the performance being judged: Since the intended usage of word complexity is in fields of summarization and simplification, the complexity of a text is assessed in terms of readability. The performance of the proposed method is compared with the Flesch Reading ease[18][19] measure for readability. The Flesch Reading Ease is used as a benchmark. Only the correlation is measured since the goal of the program is to be able to pick a less complex word between two given words.

Which data is being used: For this experiment, articles from The New York Times [20] are considered. This source is chosen because the articles are known for having a more refined and sophisticated written literature in terms of both structure and vocabulary.

How the articles are selected: The top 10 trending articles from NYT are chosen. The articles are chosen without logging in to the site to avoid bias.

Procedure: ***Flesch Reading ease***[18] is a standard used for judging the readability of a text. It is used here to correlate the complexity score of the algorithm with. Ideally, the algorithm output and the Flesch Reading ease should be inversely correlated since higher the reading ease, lower the complexity of the text. Their readability is checked on the Flesch Reading Ease and noted. Next, the articles are run through the proposed algorithm to measure complexity. The graph for Flesch Reading ease and the complexity score of the program is then plotted as shown in **Figure 1.** The algorithm is also tested on a manually classified lists of words : GRE easy and advanced lists of Manhattan Prep[21]. 100 random easy words and their 100 difficult synonyms are picked and run through the algorithm. **Figure 2** shows the plot of their scores which are discussed in section 4. Next, the results and observations of the test are discussed.

4. OBSERVATIONS

The goal of the algorithm is to make it easier to simplify texts, but for that it must first be verified if it proves to be an indicator of complexity. The first test, testing correlation with the Flesch Reading Ease scale does exactly that. 10 Articles of NewYork Times are taken and their reading ease and complexity scores are measured. The results are depicted in **Table 1.** As it can be seen in **Figure 1**, the correlation is strong. The values are inversely correlated since as Flesch Reading ease increases, the ease of reading is more but in our metric, as the value increases, the ease of reading decreases and the complexity increases. It is then tested for the said purpose of comparing two words: difficult words and their simpler synonyms are used. The words are randomly picked from the Manhattan 1000 wordlist for GRE which contains high level words and their simpler synonyms.

Table 1: Articles, their complexity score and reading ease

Article	complexity score(cp)	Flesch reading ease
Coding Bootcamps	4.42	64
Self Driving Trucks	5.48	45.5
Alice	3.07	78.02
The Jail Story	2.79	76.8
Afghanistan and US	5.09	55.2
GOT	3.97	63.1
Populism	4.42	45.4
Hillary Clinton	3.5	60.9
Navy	4.506	55.7

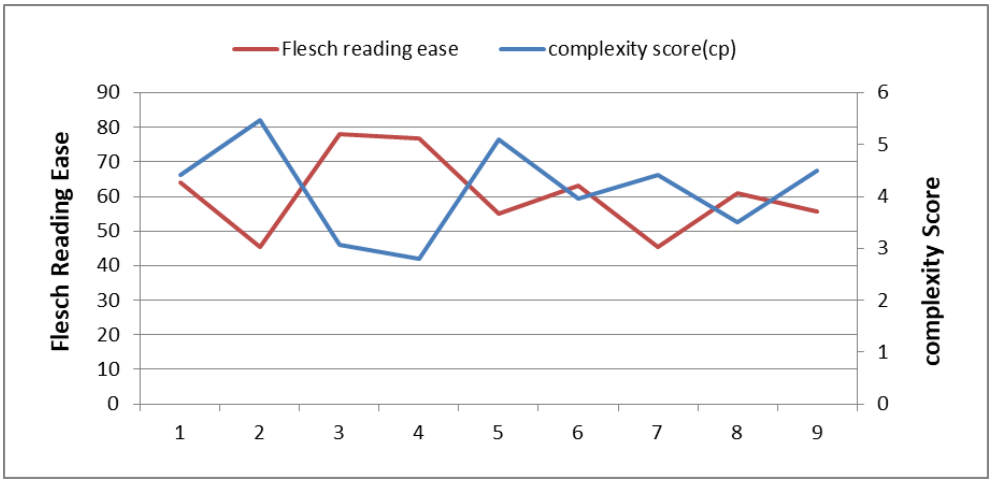

Figure 1: correlation between complexity score and reading ease

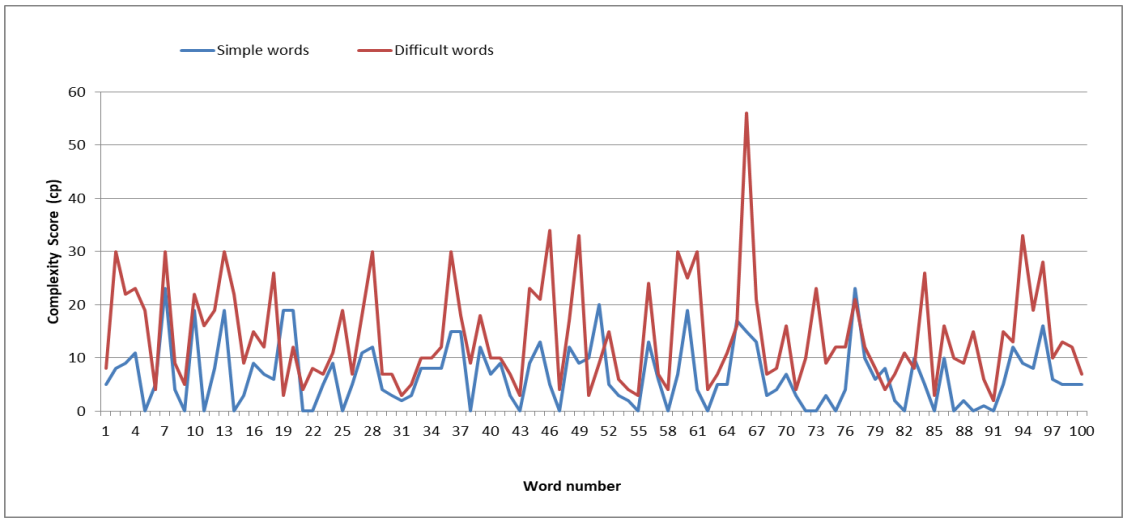

Figure 1: Complexity scores for various words and their difficult synonyms

The complexity scores for both sets are depicted in the graph in **Figure 2**. We find that there is a 93% accuracy. Out of 7 words that fail to adhere the pattern, most of them are simple words that have not been included in the basic list but should have been: deface, swell, sheer, rumour, inactivity, cheerfulness, eager, delicious The reason they give a high complexity score than their synonyms is because since they're basic words and basic words are defined more elaborately since there is no one word to define them. Hence, reaffirming the importance of basic list. The basic list might never be exhaustive, however, it can be built using dynamic methods to give good enough results.

The dynamic methods can consist of including all words form texts that have been marked simple or below the required complexity threshold and using multiple wordlists such as basic english wordlist. By these observations, it can be affirmed that the proposed method is successful by and large in determining the overall complexity of the text and the complexity of individual words.

5. USAGE

The algorithm can be tailored to the application by adjusting the following four factors:

Dictionary: The dictionary used is a huge influence on the word definition and hence the word complexity. Here we use the inbuilt python dictionary in NLTK.

Basic words: The level of complexity is adjustable, meaning, it give complexity of the text relative to that list. More higher level words in the basic words would rate high level articles as less complex as well since the high level words are assumed to be comprehensible by the user.

Internal Looping*:* What if two words are written in each other's definitions. This causes the program to go in an infinite loop. To deal with this, the number of iterations in our program is constricted to two. Various depths will give different accuracy and they can be explored based on the resources available and the accuracy required.

Which Definition. Context detection is still an open ended problem, meaning given n definitions for a word, we won't know which one to use. However, calculating the complexity and then averaging the sum does give a usable estimate.

Tweaking the above factors can provide a robust approach that can be used for any domain to generate relative simplicity. For example, in defining the complexity score for a scientific text, various terminologies can be determined which count as basic. This basic list will then be in context with the scientific documents and will be able to judge them accurately.

For simplification, a word that has the complexity score above the threshold can be chosen and it's synonyms can be

extracted from its Synset, a collection of synonyms that are interchangeable without affecting the meaning of the sentence[22]. Followed by which, the word with the lowest complexity score in the Synset can be chosen as a replacement. Further work can be done in this direction to test the efficiency of the method against existing methods. Secondly, to provide more robustness in general English, this algorithm can be paired with other methods such as word complexity using length and word complexity using usage. The three metrics can be run through a machine learning algorithm like neural networks or decision tree which would determine the weightage of each of the factors and perhaps provide an even more accurate picture of the entire text.

The algorithm doesn't perform stemming since at this scale, it is possible to work without it. However, for practical use in simplification, the words would have to be matched to the basic word list after being stemmed. Stemming would in fact increase the accuracy. Since without stemming, *'cheerful'* and *'cheerfulness'* map to different words, stemming would reduce the redundancy of the basic word list too. Following which, next step would be to match the synonyms word tense with the existing word's grammatical form and replace it

6. CONCLUSION

Semantic complexity has hence been measured using word-tree like intuition and has proven to help determine relative complexity between two words or between two articles as well. The proposed scale for semantic complexity is measured against Flesch Reading Ease and is found to have a strong inverse correlation. The method also gives positive results on hundred sets of simple words and their difficult counterparts since the complexity score is higher generally for the difficult words. This higher complexity score for difficult words validates the method and helps making simplification easier since this frees the user from requiring a ton of data and also introduces robustness and makes the system dynamic. The proposed method also preserves the meaning and the conceptual complexity of the word in a language. It can be used in text simplification for determining the simpler word of two given words, or in choosing a simpler version of the article. Apart from that, it can be used for domain specific applications such as in medicine or scientific texts to simplify texts or judge the level of complexity. Research can be done on its accuracy and usability in narrow domain scenarios. However, the algorithmic complexity for this method can still be improved. The method can also be further developed to be more accurate and scalable. It is hoped that the method will be useful in applications of simplification and carves way for other text-based semantic algorithms.

7. REFERENCES

[1] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-End Training of Deep Visuomotor Policies," vol. 17, pp. 1–40, 2015.

[2] J. Hirschberg and C. Manning, "Advances in natural language processing," *sciencemag*, vol. 349, no. 6245, pp. 394–416.

[3] T. Ingold, "Jumping NLP Curves: A Review of Natural Language Processing Research," *J. R. Anthropol. Inst.*, vol. 4, no. January, pp. 771–773, 2014.

[4] M. Shardlow, "A Survey of Automated Text Simplification," *Int. J. Adv. Comput. Sci. Appl. Spec. Issue Nat. Lang. Process.*, pp. 58–70, 2014.

[5] R. Chandrasekar and B. Srinivas, "Automatic induction of rules for text simplification," *Knowledge-Based Syst.*, vol. 10, no. 3, pp. 183–190, 1997.

[6] R. Chandrasekar, C. Doran, and B. Srinivas, "Motivations and Methods for Text Simplication," *Proc. COLING '96*, pp. 1041–1044, 1996.

[7] S. Vajjala Balakrishna, "Analyzing Text Complexity and Text Simplification: Connecting Linguistics, Processing and Educational Applications," *PhD Thesis*, vol. 1, 2015.

[8] I. Temnikova, "Text Complexity and Text Simplification in the crisis management domain," *Comput. Linguist.*, 2012.

[9] A. Siddharthan, "Syntactic simpllification and cohesion," *Tech. Report, Univ. Cambridge*, no. 597, 2004.

[10] P. Mukherjee *et al.*, "NegAIT: A new parser for medical text simplification using morphological, sentential and double negation," *J. Biomed. Inform.*, vol. 69, no. March, pp. 55–62, 2017.

[11] X. Zhang and M. Lapata, "Sentence Simplification with Deep Reinforcement Learning," 2017.

[12] M. L. Lewis and M. C. Frank, "The length of words reflects their conceptual complexity," pp. 1–42.

[13] S. Pinker, "The Language Instinct," New York, NY: Harper Perennial Modern Classics, 1994.

[14] "Ogden's Basic English Words." [Online]. Available: http://ogden.basic-english.org/words.html. [Accessed: 13-Sep-2017].

[15] "Wikipedia:List of 1000 basic words - Simple English Wikipedia, the free encyclopedia." [Online]. Available: https://simple.wikipedia.org/wiki/Wikipedia:List_of_1000_basic_words. [Accessed: 13-Sep-2017].

[16] L. Dolamic and J. Savoy, "When Stopword Lists Make the Difference," no. 1, pp. 200–203, 2009.

[17] S. Bird, "NLTK: The natural Language Toolkit," *21st Int. Conf. Comput. Linguist.*, no. July, p. 69, 2006.

[18] Rudolf Flesch, "Guide to Academic Writing Article - Management - University of Canterbury - New Zealand." [Online]. Available: http://www.mang.canterbury.ac.nz/writing_guide/writing/flesch.shtml. [Accessed: 10-Sep-2017].

[19] F. O. G. Count, F. Reading, and E. Personnel, "Derivation of new Readability formulas," 1975.

[20] "The New York Times - Breaking News, World News & Multimedia." [Online]. Available: https://www.nytimes.com/?mcubz=3. [Accessed: 13-Sep-2017].

[21] "The Manhattan Prep GRE Advantage | Comprehensive GRE Prep Books & GRE Online Study Resources | Manhattan GRE Prep." [Online]. Available: https://www.manhattanprep.com/gre/studentcenter/flash-cards/gre-flashcards.cfm. [Accessed: 07-Sep-2017].

[22] C. Fellbaum, "WordNet," *Encycl. Appl. Linguist.*, 2012.