HELLENIC REPUBLIC · IONIAN UNIVERSITY 1984

Text Simplification

Team Members:

Karageorgiou Andreas P2018122
Karvounis Alexandros P2019020
Varelas Dimitrios P2018019
Tziallas Efthimios P2018015
Baltatzidis Kyriakos P2018176
Zervakis Theodoros P2019107

Literature review:

In "Learning to Simplify Sentences Using Wikipedia" from William Coster and David Kauchak used the same data set (containing 137k aligned sentence pairs) extracted from wikipedia and they introduced a translation model for text simplification that extends a phrase-based machine translation approach.

In “Exploring Neural Text Simplification Models” from Nisioi S. , in order to train their models, they used the publicly available dataset provided by Hwang (2015) based on manual and automatic alignments between standard English Wikipedia and Simple English Wikipedia (EW–SEW). They discarded the uncategorized matches and they used only good matches and partial matches which were above the 0.45 threshold.

In “Complex Word Identification in Vietnamese: Towards Vietnamese Text Simplification” from Nguyen P. and Kauchak D., we can see that Text Simplification is a task that focuses on improving the readability and understandability of text, while preserving the original content and meaning. It is often the first step in Lexical Simplification (Shardlow, 2013) and has applications for a wide range of human and non-human audiences.

In “Improving Text Simplification Language Modeling Using Unsimplified Text Data”, Kauchak D. explains that text simplification is a monolingual translation task where text can be used from both the input and output domain to train a language model. A combined model using both simplified and normal English text achieves a 23% improvement on the lexical simplification task over a model trained only on simple data.

Basic Terms

Text simplification is the process of modifying written text to make it easier for the reader to understand. This is achieved by reducing complexity and ambiguity, using simple language, short sentences and familiar words. Text simplification is beneficial for people with reading difficulties, such as dyslexia, or for those who are learning a new language. Additionally, text simplification can make information more accessible to a wider audience. The goal of text simplification is to retain the original meaning of the text while making it easier to understand. It involves techniques such as rephrasing, summarizing, and removing unnecessary information.

Machine learning models, such as neural networks, can be trained on large amounts of text data to identify patterns and relationships between words and phrases, and then generate simpler versions of text that retain the meaning of the original text. This can be achieved by techniques such as lexical substitution, sentence compression, and text rewriting. The use of machine learning in text simplification has the potential to greatly enhance the accessibility of written content and improve literacy rates, especially in developing countries.

Text complexity is a measure of the level of difficulty or sophistication involved in comprehending written text. It encompasses various factors such as vocabulary, sentence structure, and context, as well as the reader's background knowledge and experience. Text complexity affects the ease with which a reader can understand the meaning of the text and can influence factors such as reading speed, comprehension, and motivation. Determining text complexity is important for a variety of purposes, including educational assessment, text classification, and accessibility for individuals with different levels of language proficiency. To assess text complexity, researchers and practitioners have developed various tools and metrics, including readability formulas, vocabulary analysis, and comprehension tests. By understanding text complexity, it is possible to create written materials that are accessible to a wide range of readers, promoting education and literacy for all.

Now that we have established the basics, we can analyze our review more.

Attribute Selection

After attribute selection we are left with 11 features instead of the initial 15 features which are the following: Flesch Kincaid, Reading Ease, Difficult Words, Linsear Formula, Dale-Chall, Autoread, Gunning Fog, McAlpine, Spache Read, time that takes to read the sentence, number of letters per sentence.

We performed attribute selection using CFS Subset evaluator and Best First as search method. It decreased model’s build time, but also decreased accuracy of the model by an average of 1% with each classifier.

Partition Membership

In WEKA, partition membership refers to the division of a dataset into multiple partitions or subsets for the purpose of cross-validation or model evaluation. These partitions are used to train and test machine learning algorithms to assess their performance, reduce overfitting, and improve model accuracy. Each instance in the dataset is assigned to one of the partitions, typically with roughly equal number of instances in each partition.

Dataset

The dataset we chose to get our sentences from, contains 137K aligned sentence pairs. Those sentences have a similarity above 0.50. The original version of the dataset was created out of Wikipedia pages in 2010 and then William Coster and David Kauchak generated this dataset in 2011. William Coster and David Kauchak generated a parallel simplification corpus by aligning sentences between English Wikipedia and Simple English Wikipedia in order to make the dataset eligible for text simplification and machine learning corpus as well as Weka related projects (.arff)

Feature Analysis

1) Flesch Kincaid Grade Level: The Flesch Kincaid Grade Level is a widely used readability formula which assesses the approximate reading grade level of a text. If a text has a Flesch Kincaid level of 8, this means the reader needs a grade 8 level of reading or above to understand it. Even if they're an advanced reader, it means the content is less time-consuming to read.

Mathematical Type: 0.39 (total words/total sentences) + 11.8 (total syllables/total sentences) – 15.59.

2) Flesch Reading Ease: The Flesch Reading Ease scores a text between 1 and 100, with 100 being the maximum readability score. A score between 70 and 80 is equivalent to a school grade of 8. This means that the text should be easy enough for an average adult to read.

Mathematical Type: 206.835-1.015 (total words/total sentences)-86.4(total syllables/total sentences).

3) Difficult words: By using the Twinword tool, we understand that the approach to the specific issue has to do with the frequency of occurrence of the words, thus evaluating the respective vocabulary. The language database is updated frequently so that the results are as accurate as possible. To make it easier, there is a rating system divided into 10 levels of difficulty. So this tool takes a word or text as input and displays the difficulty level from 1 to 10 (or 0 to 1 to be more understandable).

4) Monosyllabic Count: Words like 'cat' and 'jump' are monosyllabic. They contain only one syllable and are extremely interesting both in the study of their phonology (study of the sound of a language) and in the study of their morphology (study of word structure). Monosyllabic words can be of two types. Either verbal (those that describe an object or an action), or grammatical (words like 'the' and 'of' that are used without any grammatical meaning).

5) Syllable Count: Syllable Count is the number of syllables in a word, sentence, or text, which helps gauge the language complexity. It reveals the language proficiency of the speaker, whether a child, student, or language learner. For example, a child starting with simple words like "mummy" or "cat" and gradually using more complex words like "football" indicates an increase in language complexity.

6) Lexicon Count: In Lexicon-based sentiment analysis, a text message is represented as a bag of words and sentiment values from a pre-made sentiment lexicon are assigned to positive and negative words/phrases. The sentiment score is calculated by aggregating the sentiment values of all the words in the message. A lexicon is a dictionary containing information about words or word strings and can cover a language or subject area's vocabulary.

7) Linsear Write: The Lensear Write (formula) calculation was developed by the US Air Force so that aircraft technical instructions could be understood. It is based on manuals and notes that have three or more syllables.

Mathematical Type: The standard Linsear Write metric Lw runs on a 100-word sample:

1. For each "easy word", defined as words with 2 syllables or less, add 1 point.
2. For each "hard word", defined as words with 3 syllables or more, add 3 points.
3. Divide the points by the number of sentences in the 100-word sample.
4. Adjust the provisional result r:
   - If $r > 20$, $Lw=r/2$
   - If $r \leq 20$, $Lw=r/2 - 1$

The result is a "grade level" measure, reflecting the estimated years of education needed to read the text fluently.

8) Dale–Chall: The Dale–Chall criterion refers to the difficulty readers experience when reading a text. It uses a list of 3000 words that groups of fourth graders in an American school can satisfactorily understand. At the same time, any word that is not on the list is considered difficult.

Mathematical Type: 0.1579(difficult words/words)(100)+0.0496(words/sentences).

If the percentage of difficult words is above 5%, then add 3.6365 to the raw score to get the adjusted score, otherwise the adjusted score is equal to the raw score. Difficult words are all words that are not on the word list, but it has to be considered that the word list contains the basic forms of e.g. verbs and nouns. Regular plurals of nouns, regular past tense forms, progressive forms of verbs etc have to be added.

9) Automated Readability Index: We count the number of input sentences. We measure the average length of the input sentences (number of words). We count the average number of syllables of each word in the input. We count the average number of characters of each word in the input. The final result reflects how easy a text is to read.

Mathematical Type: 4.71 x (characters/words) + 0.5 x (words/sentences) – 21.43.

10) Gunning Fog: The Gunning Fog index refers to the English language and specifically to the number of years of formal education someone would need to understand a text with their first reading. For example, a grade level index of 12 would require an age of approximately 18 years old.

Mathematical Type: Readability score = 0.4 * ((words per sentence) + (100 * (complex words / total words))).

11) McAlpine Eflaw: When referring to a global audience, we want the text to be understandable by people whose native language is not necessarily English. So Rachel McAlpine wrote an article with tips on how to make texts easy to read and easy to translate. Thus, he introduced the McAlpine EFLAW score. This number is a function of the number of words in the text, the number of small words (3 or more characters) and the number of sentences.

Mathematical Type: McAlpine Eflaw = (mini words+words)/sentences

McAlpine EFLAW(TM) Readability Score: 19.2 (very easy to understand)

12) Spache Readability Formula: The Spache Readability Formula results in the identification of the level that an average American student should have. For example, if the result is number 3, the text is understandable by an average student of 3rd grade.

Mathematical Type: Readability score = (0.141 x (words per sentence)) + (0.086 x (difficult words / total words)) – 0.42.

13) Reading Time: is based on the average reading speed of an adult (around 275 words per minute). We take the total number of words in a text and convert it into minutes.

Mathematical Type: Reading time (minutes) = (Number of characters/reading speed constant) / 60.

14) Character Count: counts the individual letters of the words in a text, as well as punctuation marks and spaces between the words.

Mathematical Type: is a simple mathematical operation that can be done by using built-in functions or manually counting the number of characters in the string.

15) Letter Count: counts only the individual letters of the words in a text, without taking into account the punctuation marks and spaces that may be present.

Mathematical Type: can be represented as a function that takes a string as input and returns the number of letters in that string as output.

16) Frequency Matrix: is a table showing the frequency of terms in a set of documents. Used in NLP and text mining, it's a rectangular table where rows are unique terms and columns are unique documents. The cells show the number of occurrences of a term in a document.

Mathematical Type: can be represented as a two-dimensional array, where the rows represent the terms and the columns represent the documents. Each element in the array represents the number of occurrences of a given term in a given document.

17) Stop words: are words that we encounter very often in sentences and make the text more difficult. They are minimally important for analyzing a text and would be preferred not to be used in the input.

Mathematical Type: can be represented as a list or set of words that is used to remove commonly used words in a language, to improve computational efficiency and to focus on keywords that are most informative.

Experimental Procedure and Results

Table for dataset with 8K words

Classifier	Correctly classified(%)	TP Rate	FP Rate	Precision	Recall	F-measure
J48	59.8221	0.598	0.402	0.600	0.598	0.597
J48/AttribSel	60.1264	0.601	0.399	0.603	0.601	0.600
J48/ Partition	64.0716	0.641	0.359	0.647	0.641	0.637
J48/AttribSel/ Partition	63.6156	0.636	0.364	0.639	0.636	0.634
IBK	42.6949	0.427	0.573	0.426	0.427	0.426
IBK/AttribSel	42.6598	0.427	0.573	0.426	0.426	0.425
IBK/ Partition	64.6219	0.646	0.354	0.652	0.646	0.643
IBK/AttribSel /Partition	63.8375	0.638	0.362	0.641	0.638	0.637

Classifier	Correctly classified(%)	TP Rate	FP Rate	Precision	Recall	F-measure
Classification via Regression	61.2854	0.613	0.387	0.614	0.613	0.612
Classification via Regression/AttribSel	60.9108	0.609	0.391	0.610	0.609	0.608
Classification via Regression/Partition	64.0716	0.641	0.359	0.645	0.641	0.638
Classification via Regression/AttribSel/Partition	62.8073	0.628	0.372	0.631	0.628	0.626

Table for dataset with 145K words

Classifier	Correctly classified(%)	TP Rate	FP Rate	Precision	Recall	F-measure
J48	61.7885	0.618	0.382	0.618	0.618	0.617
J48/AttribSel	61.8967	0.619	0.381	0.620	0.619	0.618
J48/ Partition	63.3652	0.634	0.366	0.635	0.634	0.633
J48/AttribSel/ Partition	63.042	0.630	0.370	0.633	0.630	0.629
IBK	46.167	0.462	0.538	0.461	0.462	0.460
IBK/AttribSel	46.7045	0.467	0.533	0.466	0.467	0.464
IBK/ Partition	63.6216	0,636	0,364	0,638	0,636	0,635
IBK/AttribSel /Partition	63.144	0,631	0,369	0,633	0,631	0,630

Classifier	Correctly classified(%)	TP Rate	FP Rate	Precision	Recall	F-measure
Classification via Regression	62.2109	0.622	0.378	0.623	0.622	0.621
Classification via Regression/AttribSel	62.0917	0.621	0.379	0.623	0.621	0.620
Classification via Regression/Partition	—---	—---	—---	—---	—---	—---
Classification via Regression/Partition/Partition	62.6506	0.627	0.373	0.627	0.627	0.626

Result Evaluation

In this machine learning experiment a dataset of 249998 sentences is divided into two classes: simple and unsimplified sentences. The text is preprocessed using the Python NLTK library and "English pickle" to split the text into sentences. The Text Stat library is used to calculate 15 different text statistics that serve as features for the model. The features are then used to train three different machine learning algorithms: classification via regression, IBk and J48. The results are evaluated using 10-fold cross-validation, a method that involves dividing the dataset into 10 equal parts, using 9 parts for training and 1 part for testing, repeated 10 times. The average performance metric across all iterations is used as the final evaluation of the model's performance. This method balances the number of samples used for training and testing while also reducing the risk of overfitting. We experimented using 2 different train samples (8542, 145108) to test if train sample size actually has an impact on solving the problem. We also investigated if Partition Membership and Attribute selection impacts the accuracy of the model.

Attribute selection decreased the build time of the model but also improved the models accuracy at IBK, J48 classifiers except CvR which actually decreased its performance due to the nature of the CvR classifier. But combining Attribute Selection and Partition Membership yield lower accuracy than using Partition Membership only. Time on the other side, increased significantly when we used Partition Membership. So preprocessing the dataset impacts our model's accuracy.

The results are shown above in detail and below there is a preview of our average statistics.

Dataset	8K	145K
Correctly Classified	64.6219	63.6216
Avg Precision	0.633	0.591
Avg Recall	0.592	0.593
Avg F-Measure	0.628	0.585
Avg TP-Rate	0.592	0.594
Avg FP-Rate	0.407	0.406

After thorough research with similar references, results based on the same dataset we used have shown that the average precision is 69-70% (1), compared to our 64%. One of the reasons it might differ is the fact that we used slightly more than half the dataset, more specifically 145k out of 250k words.

Conclusion and Future Improvements

In this paper, we have conducted a literature review about text simplification. In our review, we used a dataset taken from Wikipedia and adapted it for the purpose of text simplification. In simple English, what we did was get a dataset divided into simple and complex sentences, add features in order to collect data based on every sentence and then run Machine learning algorithms with the data we had aiming to teach an algorithm to recognise if a sentence is simple or complex. Our best result is 64,6% with the algorithm IBk in a 10 cross validation and then algorithms such as J48 and Classification via Regression with results varying from 61-63%. It is important to note that those algorithms needed much more time to classify compared to other “lighter” algorithms.

These results are obviously lower than what we obtained from the reference materials, but bearing in mind that our project had a timeline and we didn't have the freedom to run more algorithms, features and of course the equipment to execute bigger datasets without crashing.

As for future improvements, we could have taken a bigger dataset with more valuable sentences in order to have more accurate results. Another way we can improve the model accuracy is by using more effective data for training or incorporating sentence structure, TF-IDF and other factors like word co-occurrence.

The *diversity* between our experiments and the ones to the references, is that we used Attribute selection and Partition Membership before to classify our algorithms in order to get the best possible results .

References

Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, et al.. ASSET: A Dataset for Tuning and Evaluation of Sentence Simplification Models with Multiple Rewriting Transformations. ACL 2020 - 58th Annual Meeting of the Association for Computational Linguistics, Jul 2020, Seattle / Virtual, United States. ffhal-02889823f

Kauchak, D. and Coster, W. (2011) Text simplification data sets. Available at: https://cs.pomona.edu/~dkauchak/simplification/.

Nguyen, P. and Kauchak, D. (2022) “Complex word identification in Vietnamese: Towards Vietnamese text simplification,” Proceedings of the Workshop on Multilingual Information Access (MIA) [Preprint]. Available at: https://doi.org/10.18653/v1/2022.mia-1.6.

Pavlick, E. and Callison-Burch, C. (2016) “Simple PPDB: A paraphrase database for simplification,” *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* [Preprint]. Available at: https://doi.org/10.18653/v1/p16-2024.

Nisioi, S. *et al.* (2017) “Exploring neural text simplification models”, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* [Preprint]. Available at: https://doi.org/10.18653/v1/p17-2014.

Coster, W. and Kauchak, D. (no date) - aclanthology.org, Learning to Simplify Sentences Using Wikipedia. Available at: https://aclanthology.org/W11-1601.pdf

Coster, W. and Kauchak, D. (no date) Simple english wikipedia: A new text simplification task - pomona. Available at: https://cs.pomona.edu/~dkauchak/papers/kauchak11simple.short.pdf

Kauchak, D. (no date) Improving text simplification language modeling using UNSIMPLIFIED text data, ACL Anthology. Available at: https://aclanthology.org/P13-1151/ .