Text Simplification for Enhanced Readability*

Siddhartha Banerjee, Nitin Kumar and C. E. Veni Madhavan
Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India

Keywords: Mining Text and Semi-structured Data, Text Summarization, Text Simplification.

Abstract: Our goal is to perform automatic *simplification* of a piece of text to enhance readability. We combine the two processes of *summarization* and *simplification* on the input text to effect improvement. We mimic the human acts of incremental learning and progressive refinement. The steps are based on: (i) phrasal units in the parse tree, yield clues (handles) on paraphrasing at a local word/phrase level for simplification, (ii) phrasal units also provide the means for extracting segments of a prototype summary, (iii) dependency lists provide the coherence structures for refining a prototypical summary. A validation and evaluation of a paraphrased text can be carried out by two methodologies: (a) standardized systems of readability, precision and recall measures, (b) human assessments. Our position is that a combined paraphrasing as above, both at lexical (word or phrase) level and a linguistic-semantic (parse tree, dependencies) level, would lead to better readability scores than either approach performed separately.

1 INTRODUCTION

Many knowledge-rich tasks in natural language processing require the input text to be easily readable by machines. Typical tasks are question-answering, summarization and machine translation. The representation of information and knowledge as structured English sentences has several virtues, such as readability and verifiability by humans. We present an algorithm for enhancing the readability of text documents.

Our work utilizes: (i) Using Wordnet (Miller et al., 1990) for sense disambiguation, providing various *adjective/noun* based relationships, (ii) ConceptNet (Liu and Singh, 2004) capturing commonsense knowledge predominantly by *adverb-verb* relations; (iii) additional meta-information based on grammatical structures. (iv) Word rating based on a classification system exploiting word frequency, lexical and usage data. We use the evaluation schemes based on standard Readability metrics and ROUGE (Lin and Hovy, 2003) scores.

Our present work falls under the category of summarization with paraphrasing. With our planned incorporation of other semantic features and aspects of discourse knowledge, our system is expected to evolve into a *natural* summarization system.

The remainder of the article is structured as follows. We give an overview of related work on text simplification in section 2. We discuss our approach in Section 3. Experiments and results are presented in Section 4. Discussion on future work concludes the paper.

2 RELATED WORK

The task of text simplification has attracted a significant level of research. Since the work of Devlin *et al.* (Devlin and Tait, 1993), text simplification has received a renewed interest. Zhao *et al.* (Zhao et al., 2007) aim at acquiring context specific lexical paraphrases. They obtain a rephrasing of a word depending on the specific sentence in which it occurs. For this they include two web mining stages namely *candidate paraphrase extraction* and *paraphrase validation*. Napoles and Dredze (Napoles and Dredze, 2010) examine Wikipedia *simple articles* looking for features that characterize a simple text. Yatskar *et al.* (Yatskar et al., 2010) develop a scheme for learning lexical simplification rules from the edit histories of simple articles from Wikipedia. Aluisio *et al.* (Aluisio et al., 2010) follow the approach of tokenizing the text followed by identification of words that are complex. They identify the complexity of words based on a compiled Portuguese dictionary of

*Work supported in part under the DST, GoI project on Cognitive Sciences at IISc.

Banerjee S., Kumar N. and E. Veni Madhavan C..
Text Simplification for Enhanced Readability.
DOI: 10.5220/0004626102020207
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval and the International Conference on Knowledge Management and Information Sharing* (KDIR-2013), pages 202-207
ISBN: 978-989-8565-75-4

simple words. For resolving part-of-speech (POS) ambiguity they use MXPOST POS tagger. Biran *et al.* (Biran et al., 2011) rely on the complex english Wikipedia and simple english Wikipedia for extraction of rules corresponding to single words. The replacement of a candidate word is based on the extracted context vector rules.

Our approach differs from the previous approaches in that we combine the syntactic and semantic processing stages with a calibrated word/phrase replacement strategy. We consider individual words as well as n-grams as possible candidates for simplification. Further we achieve the task of simplification in two phases: (i) identification of complex lexical entities based on a significant number of features, (ii) replacement based on sentence context. Based on the efficacy of this set up, we propose to expand the constituents to include phrasal verbs, phrasal nouns and idioms.

3 SUMMARIZATION AND SIMPLIFICATION

The simplification task addresses issues of readability and linguistic complexity. The basic idea of our work is to segment the parse tree into subtrees guided by the dependency complexity and then rephrase certain complex terminal words of subtrees with similar, simpler words. Lexical simplification substitutes the complex entities with simple alternatives. Syntactic simplification fragments large and complex sentential constructs into simple sentences with one or two predicates. We present an outline of the stages of the algorithm in Sections 3.1, 3.2, the steps of the algorithm in Section 3.3 and an example in Section 3.4.

3.1 Lexical Simplification

The Lexical Simplification problem is that of substituting difficult words with easier alternatives. From technical perspective this task is somewhat similar to the task of paraphrasing and lexical substitution. The approach consists of the following two phases:

1. *Difficult Words Identification.* In the first phase we take a set of 280,000 words from the Moby Project (Ward, 2011) and a standard set of 20,000 words ranked according to usage from a freely available Wikipedia list (WikiRanks, 2006). The first set contains many compound words, technical words, and rarely used words. The latter list contains almost all the commonly used words (about 5,000) besides the low frequency words. This list has been compiled based on a corpora of about a billion words from the Project Gutenberg files.

For the set of *difficult* words, for the baseline we pick the least ranked words from the Wikipedia list and other words from the Moby corpus totalling upto 50,000 words. For the set of *simple* words, we use the Simple English Wikipedia. To quantify the relative complexity of these words we consider the following features:

- *Frequency*. The number of times a word/bigram occurs in the corpus under consideration. We consider the frequency as being inversely proportional to the difficulty of a word. For computing the frequency of a word we used the Brown corpora (Francis and Kucera, 1979) which contains more than one million words.
- *Length*. We consider the number of letters the word contains. We consider that with increase in length the lexical complexity of a word increases.
- *Usage.* This feature is measured in terms of the number of senses the n-gram has ($n \leq 3$). For another measure of complexity we use the number of senses reflecting diverse usage. This kind of usage pattern in turn reflects commonness or rareness. We use the Wordnet for the number of senses.
- *Number of Syllables.* A syllable is a segment of a word that is pronounced in one uninterrupted sound. Words contain multiple syllables. This feature measures the influence of number of syllables on the difficulty level of a word. *Number of Syllables* are obtained using the CMU dictionary (Weidi, 1998). If the syllable breakup of a given word is not found in the CMU dictionary we use standard algorithm based on vowel counts and positions. We consider the number of syllables as being proportional to the difficulty level.
- *Recency.* Words which have been newly added are less known. We obtain sets of such words by comparing different versions of WordNet and some linguistic corpora from news and other sources. In our case *Recency* feature of a word is obtained by considering the previous 13 years of compilation of recent words maintained by OED website. We use a boolean value based on word being recent or not according to this criterion.

Considering the above features normalized scores are computed for every word of the two sets of difficult and easy words. We use this data as training sets for an SVM (Cortes and Vapnik,) based classifier. The results of classification experiments are presented in Table 1. We get an overall accuracy over 86%.

2. *Simplification.* In this stage the above features are used to generate rules of the form:

   $$<\textit{difficult} \Longrightarrow \textit{simple}>$$

   e.g. $<$*Harbinger* $\Longrightarrow$ *Forerunner*$>$, Our proposed system then determines rules to be applied based on contextual information. For ensuring the grammatical consistency of the rule $<$*difficult* $\Longrightarrow$ *simplified*$>$ we consider only those *simplified* forms whose POS-tags match with the *difficult* form. For this we create a POS-tag dictionary using the Brown corpora in which we store all possible POS-tags of every word found in the corpora.

3.2 Syntactic Simplification

We mention the tools and techniques used for the intermediate stages. We use the Stanford Parser (Klein and Manning, 2003) based on probabilistic context free grammar to obtain the sentence phrase based chunk parses. Predicate, role and argument dependencies are elicited by the Stanford Dependency Parser. Our algorithm identifies the distinct parts in the sentence by counting the number of cross arcs in the dependency graph. Subsequently, the algorithm scans the sentence from left to right and identifies the first predicate verb. Using that verb and ConceptNet relations, the algorithm identifies the most probable sentence structure for it, and regenerates a simple representation for the sentence segment processed so far. The algorithm then marks that verb and continues the same process for the next verb predicate. This process continues till the whole sentence has been scanned.

Once all the sentences are scanned and rephrased we reorder them. We devise a novel strategy to order the rephrased sentences so as to achieve maximal coherence and cohesion. The basic idea is to reduce the dependence between the entities without disrupting the meaning and discourse. We try to find regressive arcs in the dependency graph guided by the phrase and chunk handles. This approach has the advantage of keeping a partial knowledge based approach, which allows the simplification of the syntactic structure and create a knowledge based structure in natural language. We also used the ConceptNet data to establish similarities based on analogies and used WordNet for lexical similarity.

We include some additional meta-information based on grammatical structures.

3.3 Algorithm

Now we present the steps of the algorithm in detail.

Algorithm 1: Text Simplification Abstract

1. Pre-processing.
   (a) split the document into sentences and sentences into words.
   (b) perform chunk and dependency parsing using the Stanford Parser.
   (c) find subtrees which are grammatically well formed.
2. Resolve the references using the Discourse representation Theory (Kamp and Reyle, 1990) methods and substitute full entity names for references.
3. Identify difficult word entities using a linear kernel SVM classifier trained with a powerful lexicon of difficult and simple words.
4. Replace the difficult word entities using replacement rules.
   (a) Consider the context of the difficult word w_k in terms of the wordnet senses of words of the surrounding word in the whole sentence. We indicate this for a window of size $2i$. s_x denotes the Wordnet senses for the word x.

   $$s_{w_{k-i}}, \ldots, s_{w_k}, \ldots, s_{w_{k+i}},$$

   (b) Select the lemmas $(l_1^{w_k}, \ldots, l_j^{w_k})$ which have the same sense s_{w_k}.
   (c) Replace w_k with the lemma which is most frequent in the Wikipedia context database within the current context:

   $$s_{w_{k-i}}, \ldots, s_{w_{k-1}}, s_{w_{k+1}}, \ldots, s_{w_{k+i}}$$

5. Collect the sentences belonging to the same subject using the ConceptNet and order them optimally to reduce the dependency list complexity.
   (a) Find the dependency intersection complexity of individual sentences using the Stanford Dependency graph.
   (b) Count the number of pertinent regressive arcs.
   (c) Splice the parse tree at places where longer regressive arcs appear in dependency list. Longer arcs signifies a long range dependencies due to anaphora, named entity, adverbial modifiers, co-references, main, axillary and sub-ordinate verb phrases are used as chunking handles for reordering and paraphrasing.

6. Use the generative grammar for sentence generation from the resultant parse trees. We have used the MontyLingua based language generator, modified with additional rules for better performance and output quality.
7. Rank the sentences using the top-level information from Topic Models for highly probable and salient handles for extraction. These handles point to the markers in the given sentence from the input documents for focusing attention. Select the 10% top scored sentences.
8. Reorder the sentences to reduce the dependency list complexity, utilizing information from the corresponding parse-tree.

We used a 6 GB dump of English Wikipedia[2] articles and processed it to build a context information database in the following way.

Algorithm 2: Wikipedia Context preprocessing

```
1. for each word $w$ (excluding stop-words) in every sentence of the articles, create a context set $S_w$ containing the words surrounding $w$.
   (a) add $S_w$ to the database with $w$ as the keyword.
   (b) if $w$ already exists, update the database with the context set $S_w$.
```

3.4 Example

We tested our algorithm on a sample of 30 documents from the *DUC 2001 Corpus* of about 800 words each. An extract from the document *d04a/FT923-5089* is reproduced below.

> *There are growing signs that Hurricane Andrew, unwelcome as it was for the devastated inhabitants of Florida and Louisiana, may in the end do no harm to the re-election campaign of President George Bush.*

The parse for this sentence is given in Figure 1, and the dependency is given in Figure 2. The original sentence has now been split into three sentences by identifying the cohesive subtrees and reordering them in accordance with our algorithm. Some candidate words for lexical simplification are *devastated, inhabitants*. The present output does not reflect the possibilities for rephrasing these words although lexically simpler words for replacement exist in our databases. Certain technical deficiencies in our software in this connection are presently being rectified.

The system output is given below:

> *Hurricane Andrew was unwelcome. Unwelcome for the devastated inhabitants of Florida and Louisiana. In the end did no harm to the re-election campaign of President George Bush.*

4 EXPERIMENTS AND RESULTS

In the phase involving identification of difficult words, we used the Support Vector Machine (SVM) and conducted the experiment using trained data set of sizes 1100, 5500, 11000 words where 10:1 ratio of difficult labeled and simple labeled words were used for training. The SVM reported an accuracy of more than 86% (Table 1). The overloading of the training set with difficult word samples was consciously made in view of the fact that by Zipf's law a large number of words are rarely used and hence corpus data do not provide enough samples, but these words get used sporadically. Many experiments for training with varying numbers of pre-identified set of commonly used words by rank can be conducted as follows. For the baseline we have fixed the first $n = 20,000$ ranked words as simple. Without loss of generality, n can be increased and the training runs repeated. The intuition behind fixing the ratio to 10:1 for our experiments is that the first 20,000 ranked words (which we considered "simple") account for more than 90% usage among the over 200,000 words in our set of words.

In the simplification phase, we computed the euclidean length of the vectors corresponding to both *difficult* and *simple* words, considering them as points in five dimensional space. We further hypothesized that increase in euclidean length of vectors indicated increase in complexity of words. Hence we filtered out those rules in which the euclidean length of a *difficult* word was less than euclidean length of *simple*.

For the verification of our hypothesis we conducted the experiment on different subsets of Brown corpora involving 100000, 400000, 700000 and 1100000 words and found it to be consistent. This was followed by the task involving two possibilities, first where the *difficult* word will have only one sense. As there exists only one sense for *difficult* word, irrespective of the context the word is replaced by *simple* equivalent word. In the second possibility we found the sentence context of the *difficult* word and the sentence context of all possible *simple* words from processed Wikipedia. We consider the context of *simple* word which was found to be the best match for the context in terms of intersection of words.

We evaluated the mean readability measures of the original document and the system generated out-

[2] http:/ dumps.wikipedia.org/enwiki

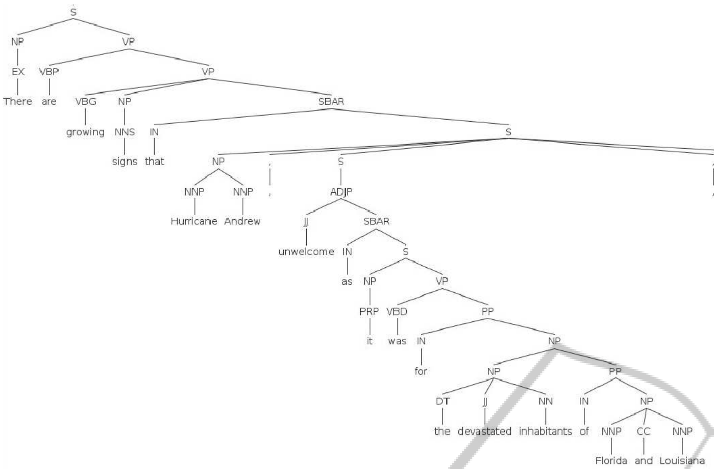

Figure 1: Parse Tree.

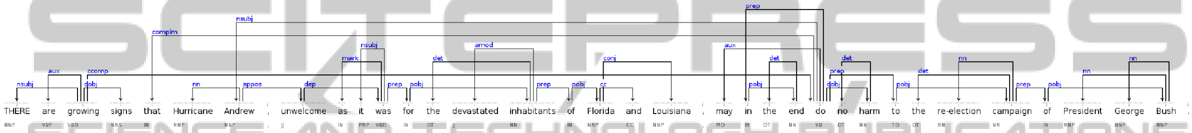

Figure 2: Dependency parse.

Table 1: Accuracy of hard word classification using SVM for different training sample sizes.

Train data / Test data	1100	5500	11000
200	90.50	88.00	88.50
400	89.75	87.25	87.75
600	90.00	88.00	88.33
800	88.62	86.50	86.75
1000	88.00	86.10	86.20
1500	88.88	87.20	87.30
2000	88.75	87.65	87.70
4000	88.65	87.90	87.92

put. The results are reproduced in the Table 2. The readability formulas use parameters like average word length, average sentence length, number of hard words and number of syllables. The paper (Vadlapudi and Katragadda, 2010) gives this set of eight most commonly used readability formulae developed by various authors for assessing the ease of understanding by students. A significant improvement is observed in all the readability measures. We note that in the first measure (FRES) the dominant terms have coefficients with negative signs as supposed to the other measures using similar terms. Further it has a large positive constant term to get the final score as a positive value. Hence the net score of FRES is the only measure that decreases with hardness.

This fact is in support of the centering theory hypothesis which we applied for splitting the sentences.

indent We used the summarization evaluation metric system ROUGE (Lin and Hovy, 2003) for ascertaining semantics and pragmatics integrity. ROUGE is a set of metrics and a software package, used for evaluating automatic summarization and machine translation software in natural language processing. The ROUGE measures are based on a comparison of frequencies of common subsequences between machine the reference and generated summaries. We computed the ROUGE scores between the original documents and the the system generated output to identify the closeness of the rephrased output with the input. The results are given in Table 3. The results are quite promising.

5 CONCLUSIONS

We have shown effective ways of segmenting and reordering sentences to improve readability. We plan to perform human centric experiments involving assessment of readability by human subjects. Further work is needed to develop a appropriate evaluation metrics which take into account structural (ROUGE, readability), grammatical and stylistic features.

Table 2: Average readability scores comparison between source sentences and our system generated output. Reference values for “upto secondary level” and upto “graduate level” have been provided for comparison.

Criterion	Source	System	Reference Values	
		output	Normal	Hard
Flesch Reading Ease	35.47	38.01	100 .. 60	30
Flesch-Kincaid Grade Level	16.76	11.14	-1 .. 16	17
RIX Readability Index	10	6	0.2 .. 6.2	7.2
Coleman Liau Index	16.05	12.25	2 .. 14	16
Gunning Fog Index	18.31	13.88	8 ..18	19.2
Automated Readability Index	18.70	12.94	0 .. 12	14
Simple Measure of Gobbledygook	13.95	12.49	0 .. 10	16
LIX (Laesbarhedsindex)	63.41	55.88	0 .. 44	55

Table 3: ROUGE scores. Average recall scores of the system generated text with the original source documents.

ROUGE	1	2	4	L	SU4	W
Avg. scores	40.3	8.0	14.2	40.3	8.0	14.2

REFERENCES

Aluisio, S., Specia, L., Gasperin, C., and Scarton, C. (2010). Readability assessment for text simplification. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–9. Association for Computational Linguistics.

Biran, O., Brody, S., and Elhadad, N. (2011). Putting it simply: a context-aware approach to lexical simplification. In *the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 496–501.

Cortes, C. and Vapnik, V. Support-vector networks. *Machine Learning*, 20(1).

Devlin, S. and Tait, J. (1993). The use of a psycho-linguistic database in the simplification of text for aphasic readers. pages 161–173.

Francis, W. and Kucera, H. (1979). Brown corpus manual. http://khnt.hit.uib.no/icame/manuals/brown/INDEX.HTM.

Kamp, H. and Reyle, U. (1990). *From Discourse to Logic/ An Introduction to the Modeltheoritic Semantics of Natural Language*. Kluwer, Dordrecht.

Klein, D. and Manning, C. D. (2003). Accurate unlexicalized parsing. In *41st annual meeting of the Association of Computational Linguistics*. ACL.

Lin, C.-Y. and Hovy, E. H. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Language Technology Conference (HLT-NAACL)*. ACL.

Liu, H. and Singh, P. (2004). Conceptnet: A practical commonsense reasoning toolkit.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. J. (1990). Introduction to wordnet: An online lexical database. *International journal of lexicography*, 3(4):235–244.

Napoles, C. and Dredze, M. (2010). Learning simple wikipedia: A cogitation in ascertaining abecedarian language. In *Proceedings of HLT/NAACL Workshop on Computation Linguistics and Writing*.

Vadlapudi, R. and Katragadda, R. (2010). Quality evaluation of grammaticality of summaries. In *11th Intl. conference on Computational Linguistics and Intelligent Text*.

Ward, G. (2011). Moby project. *http:// www.gutenberg.org/dirs/etext02*.

Weidi, R. (1998). The cmu pronunciation dictionary. *release 0.6*.

WikiRanks (2006). Wiktionary: Frequency list. *http:// en.wiktionary.org/wiki/Wiktionary:Frequency_lists*.

Yatskar, M., Pang, B., Danescu-Niculescu-Mizil, C., and Lee, L. (2010). For the sake of simplicity: Unsupervised extraction of lexical simplifications from wikipedia. *arXiv preprint arXiv:1008.1986*.

Zhao, S., Liu, T., Yuan, X., Li, S., and Zhang, Y. (2007). Automatic acquisition of context-specific lexical paraphrases. In *Proceedings of IJCAI*, volume 1794.