Neural Versus Non-Neural Text Simplification: A Case Study

Islam Nassar[†*] **Michelle Ananda-Rajah**[†‡*] **Gholamreza Haffari**[†*]
†Faculty of Information Technology, Monash University, VIC, Australia.
‡ Department of Infectious Diseases, The Alfred Hospital and Central Clinical School.
*{firstname.lastname}@monash.edu

Abstract

We propose a modular rule-based system for Text Simplification and show that it outperforms the state-of-the-art neural-based simplification system in terms of simplicity. We compare the output of both systems to highlight the differences between the two approaches. Further, we present an adaptation of our system to handle domain-specific tasks, where we employ a hybrid approach of our rule-based system and phrase-based machine translation to simplify medical discharge summaries in a low-resource situation. We compile a small medical simplification dataset to evaluate our proposed solution.

1 Introduction

Text Simplification is loosely defined as reducing the linguistic complexity of text, without changing its meaning, to suit a wider range of audience such as: non-native speakers, children, or people with language impairments. It is usually achieved by applying rewrite rules to perform two types of operations: (1) lexical simplification, where difficult words are substituted with more common alternatives; and (2) syntactic simplification, where complex sentence structures are split, reordered, or deleted to produce simpler more readable structure. To implement those rewrite rules, researchers employ various methods broadly categorized into two categories: rule-based methods and data-driven methods (Siddharthan, 2014).

In rule-based methods, the rules are hand-crafted a priori then applied to new text at simplification time. Examples of such rules include dictionary-based lookups for lexical simplification (Kurohashi & Sakai, 1999) or rules aiming at sentence restructuring into more readable formats. (A. Siddharthan, 2002; Vickrey & Koller, 2008). In contrast, data-driven methods frame the simplification process as a monolingual Machine Translation problem where the rewrite rules are learned from a parallel corpus of complex-simple sentences. This enables researchers to leverage the advances in Machine Translation to address the simplification task.

Considering the breakthrough achieved by Neural Machine Translation, we ask the questions: Could similar success be achieved in Text Simplification by employing neural architectures? Would rule-based methods be more effective since simplification is a fundamentally different task than translation?

To answer these questions, first, we propose a non-neural general-purpose rule-based simplification system. We, then, show how it can be adapted to address domain-specific simplification tasks by leveraging a small parallel dataset from the target domain. Subsequently, we compare the output of our system with that of a recently proposed neural-based simplification system (Zhang & Lapata, 2017). In our study, we focus on two simplification domains: (1) general-purpose English, for which we run our tests using Wikipedia-based datasets; and (2) medical English, for which we compile a small medical parallel corpus of complex-simple pairs and use it to test our systems. We show that our rule-based system outperforms the neural system, in terms of simplicity, both qualitatively and quantitatively. Finally, we reflect on the output of both systems to pinpoint the shortcomings of each approach and encourage researchers to address them in future research.

2 Rule-Based Simplification System

In this section, we describe our proposed rule-based simplification system. It comprises two modules corresponding to the two major operations of text simplification: lexical and syntactic simplification.

2.1 Lexical Simplification

The lexical module operates in two phases: simpler synonyms extraction, and lexical substitution. In the first phase, the system builds a synonyms dictionary for all the words appearing in the input text and apply a simplicity criterion to only keep synonyms which are simpler than the original words. In the second phase, the system decides which words should be substituted with which synonyms based on the context of the original words in their sentences.

Simpler Synonyms Extraction Given an input sentence, this phase starts with tokenization and part-of-speech (PoS) tagging.[1] Subsequently, for each (word, PoS) pair, if the PoS tag corresponds to a verb, adjective or noun (except proper nouns), the word is looked up in four lexical databases to find all possible synonyms. We use WordNet, Thesaurus, paraphrase.org, and domain-specific databases[2] to ensure a comprehensive coverage. We use the PoS tag while looking up synonyms to avoid issues arising due to polysemy, words with same spelling but different meanings (consider the difference in meaning between "lead" as a verb – guide, versus "lead" as a noun – metallic element).

To preserve grammaticality, the system, then, applies morphological changes to the obtained synonyms so that the synonyms match the PoS of the original word. The changes applied include singularization or pluralization for nouns, setting superlative or comparative forms for adjectives, and tense conjugation for verbs[3].

Finally, the system selects only the synonyms which are indeed simpler than the original word. For that we apply an intuitive simplicity criterion: if the (Synonym, PoS) pair appears in a large corpus of text[4] more often than the (Word, PoS) pair, we assume that the synonym is more common and hence is a simpler alternative of the original word. By repeating the above process on all the words appearing in the text, we obtain a simpler synonyms dictionary with many possible synonyms for each word.

Lexical Substitution In this phase, the system uses the obtained synonyms dictionary in conjunction with a language model to produce a set of candidate sentences and select the simplest among them. This process happens in an iterative greedy manner. First, the (word, Pos) pairs of the input sentence are scanned sequentially and for each pair with an entry in the synonyms dictionary, a corresponding set of sentences are produced where each sentence has the word replaced with one of the possible simpler synonyms. This set is then scored using a language model and the highest scoring sentence, based on perplexity scores, is deemed to be the simplest and hence replaces the original sentence. This process is repeated till all (word, PoS) pairs of the input sentences are scanned. The last obtained sentence is the output of the lexical simplification module.

The choice of the language model is extremely important to ensure that the sentence with the highest score is indeed the simplest. To choose a suitable language model, we ran experiments using a validation set of 2000 sentences from WikiLarge corpus. We found that the best performance with respect to simplicity metrics, was obtained using language models which had been trained on a simple English corpus. This tends to encourage output sentences which are simpler and more common. The best performance was achieved using a 5-gram language model (Brown et al. 1992) trained on the Simple Wikipedia corpus[5].

2.2 Syntactic Simplification

For the syntactic simplification, we adopted an existing open source implementation - The Multilingual Syntactic Simplification Tool (MUSST) (Scarton et al., 2017) - whereby the syntactic simplification is performed on a sentence level by applying a set of general-purpose simplification rules on its dependency parse tree. Those rules implement four operations that are arguably the most useful simplification operations.

- Splitting conjoint clauses
- Splitting relative clauses
- Splitting appositive phrases
- Changing passive-voice to active-voice

To apply the above rules, the sentence is first parsed using the Stanford dependency parser

[1] We use Stanford tokenizer and PoS-tagger in *NLTK* Python library

[2] An example of an automatically extracted medical dictionary is presented in section 3 of this paper

[3] We use *Pattern* Python package for morphology changes

[4] We use *News Crawl 2013* corpus in *WMT16* Task

[5] simple.wikipedia.org

(Chen & Manning, 2014) and then three main operations are performed to achieve the final output:
Analysis: where the sentence is analyzed in search for simplification clues such as discourse markers for conjoint clauses (ex: "and" or "when"), or relative pronouns for relative clauses (ex: "who", "which").
Transformation: where the core operations are applied to transform the sentence into a simplified form. It is applied in a recursive manner until the sentence has no more simplification clues.
Generation: where the simplified sentences are reconstructed ensuring proper grammatical structure.

3 Medical Domain Adaptation

With the above proposed system architecture, our simplification system was able to efficiently handle general-purpose English simplification such as simplifying news articles or Wikipedia text. However, it struggled when trying to simplify domain-specific text such as Medical text or Financial text. This is due to the limited coverage of general-purpose dictionaries (such as Wordnet and Thesaurus) to such domains. In this section, we show how our system can be adapted to address such domain-specific applications in a low-resource setting. We present an adaptation of our system to the medical domain, where the objective is to simplify medical discharge summary reports using a very small training set of parallel complex-simple sentences from the target domain.

Recalling our system architecture, the syntactic simplification module would have no issue simplifying domain-specific text as it operates on the sentences dependency parse tree and hence is domain-agnostic. This is not the case, however, for the lexical module; since the lexical module uses dictionaries to lookup simpler alternatives, it would fail to address domain-specific jargon which is non-existent in general-purpose dictionaries. To counteract this issue, we employed a data-driven approach to enrich the lexical module dictionaries and extend its coverage to domain specificities.

Medical Dataset First, we compiled a small parallel corpus of complex-simple medical text by manually simplifying 500 sentences drawn from "General Medicine" medical summary reports.

The 500 sentences were randomly selected from a pool which included reports with the highest lexical diversity in the entire dataset. We calculate the lexical diversity as the ratio of unique word count in a report to the total length of vocabulary. This is to ensure that the selected sentences dataset captures a diverse representation of the underlying medical reports corpus. The simplification was conducted by a medical expert and was targeted to address audience of Grade 6 level on the Flesch-Kincaid scale (Kincaid et al. 1975).

Extracting Synonyms After compiling the medical dataset, we used Moses toolkit (Koehn et al. 2007) to train a phrase-based machine translation model using 450 parallel sentences (the remaining 50 sentences were held out to test the system). One of the outputs of the trained model is the PBMT phrase table, which depicts potential mappings between source (i.e. complex) and destination (i.e. simple) phrases accompanied with maximum likelihood alignment scores for each phrase mapping. We used the phrase table to extract a phrase-synonyms dictionary of medical jargon, by scanning through each source phrase and selecting the destination phrase with the highest PBMT alignment score as its synonym phrase. Finally, we used the extracted phrase mappings dictionary to complement the general-purpose dictionaries in the lexical module, proposed in section 2.1. This yielded a great improvement of 11 points on the simplicity scale, as will be shown in more details in section 6.

4 Neural Simplification Overview

Before we proceed with the systems comparison, we, first, briefly describe the neural-based approach for text simplification as proposed by (Zhang & Lapata, 2017) dubbed as DRESS (**D**eep **RE**inforcement **S**entence **S**implification).

In their method, they treat text simplification as a sequence-to-sequence modelling task. They draw inspiration from Neural Machine Translation, where they train an encoder-decoder model on a monolingual parallel corpus of complex-simple English. To further encourage a simpler output, they train their model in a reinforcement learning framework where the reward function is a weighted combination of the output sentence relevance, simplicity, and fluency. As a proxy for relevance, they use an LSTM-based sequence auto-encoder to obtain a vector representation for both the source and output sentences, the relevance reward is then defined as the cosine similarity between those two vectors. As for the fluency reward, they use an LSTM language model trained on simple sentences to obtain a normalized perplexity score

for the output sentence. For the simplicity reward, they use the SARI metric (Xu et al. 2016) which measures the n-gram overlaps between source, output and reference sentences. SARI will be further elaborated in section 5. Finally, to encourage lexical simplification, they use a separate pre-trained encoder-decoder model, trained in a non-reinforced setting on a parallel corpus of complex-simple sentences, to obtain lexical substitution probabilities based on a given source sentence. Using the latter model favors lexical simplification operations but does not take into account the fluency of the overall output. Therefore, the output of their system is determined by linearly combining the two encoder-decoder models.

5 Experimental Setup

In their study, (Zhang & Lapata, 2017) have conducted an extensive comparison between multiple competitive simplification systems. We hence use a similar experimental setup to be able to directly use their results in our comparison.

Baseline Our baseline is simply an echo system where the input complex sentence is not simplified but rather passed through as the output. This allows a first-glance evaluation of whether a comparison system has indeed yielded a simplified output.

Datasets We perform two types of testing:
(1) General-purpose Simplification: on *WikiSmall* (Zhu et al. 2010) and *WikiLarge* (Zhang & Lapata, 2017) datasets, where the latter is a superset of the former and both are collated by automatically aligning complex and simple sentences from the ordinary and simple English Wikipedia articles. We use the same test splits used in the mentioned study (100 sentences for *WikiSmall* and 354 sentences for *WikiLarge* not containing duplicates). This enables us to use their system output directly. We don't use *Newsela* dataset (Xu et al 2015), which was used in their study, as it is not publicly available.
(2) Medical Simplification: We use the held-out test set (50 sentences) from the medical dataset mentioned in section 3 to test our system. We couldn't test the DRESS system on our medical dataset due to its extremely limited size leading to non-sensible results when used to train a neural-based architecture such as DRESS. We, therefore, only compare our results with the baseline in case of medical data.

Evaluation Metrics We use two commonly used metrics, in the simplification literature, to evaluate the systems: (1) Flesch Kincaid Grade Level (FKGL; (Kincaid et al. 1975) which is measured on a corpus level to indicate the readability of the text as a function of number of words, sentences and syllables (lower values signifies more readable text); and (2) SARI (Xu et al. 2016) which indicates the goodness of a simplification by measuring the n-gram overlap of the **S**ystem output **A**gainst **R**eferences and against the **I**nput sentence. More specifically, SARI measures the average n-gram precision and recall of addition, deletion and copying operations. It, hence, rewards deletion operations when it occurs in both the output and reference sentences. Similarly, it rewards addition/substitution where words in the output appears in the reference but not the input. This implies that producing longer output sentences doesn't necessarily lead to higher SARI scores.

6 Results

Examining the results obtained in table 1, we can see that both systems do indeed produce simpler and more readable output as opposed to the original sentences. We also see that while our non-neural system outperforms the neural system in terms of simplicity, it lags in terms of readability. This indicates that the output of our proposed system correlates better with the reference simplifications, yet it is lengthier and hence harder to read. The latter observation is attributed to the fact that our system doesn't perform deletion operations. Instead, it introduces more words during splitting and hence creates longer sentences leading to higher FKGL values (i.e. worse readability). We designed our system this way considering the task of "Medical Text Simplification". In a medical context, it is not desired to delete words but rather to elaborate on abstract terminologies and hence deletion operations were not encouraged. Incorporating further deletion rules into our syntactic module shall lead to improved readability scores.

Qualitative examination of the output of the two systems (table 2, upper) shows that the non-neural system is doing a better job in terms of both lexical and syntactic simplification. The rule-based system successfully substitutes difficult words with what seems to be reasonable and easier alternatives. It also splits composite structures into simpler form. For example, the appositive phrase in example 1 (the trickster character) and the conjoint

	WikiLarge			WikiSmall			Medical		
	FKGL	SARI	Avg. words/sent	FKGL	SARI	Avg words/sent	FKGL	SARI	Avg words/sent
No Simplification	9.2	7.2	22.61	12.1	4.5	27.8	20.13	15.7	8.4
DRESS	**6.58**	37.08	16.39	**7.48**	27.48	16.7	N/A	N/A	N/A
Rule-Based	8.37	**40.42**	20.83	9.25	**28.42**	26.29	**15.26**	**26.79**	10.2

Table 1: Evaluation results on the three datasets

Input	The tarantula, the trickster character, spun a black cord and, attaching it to the ball, crawled away fast to the east, pulling on the cord with all his strength
Reference	The **tricky** tarantula spun a black **web** and attached it to the ball. **Afterwards**, it crawled away and pulled the **web** with him
DRESS	The tarantula, the trickster character, spun a black cord and, holding it to the ball
Rule-Based	The Tarantula **turn** a black **string**. And the Tarantula **connecting** it to the ball, crawled away **soon** to the East, pulling on the **string** with all his strength. The Tarantula is the trickster character.
Input	They are culturally akin to the coastal peoples of Papua New Guinea
Reference	They are **similar** to the coastal peoples of Papua New Guinea
DRESS	They are culturally **referring** to the coastal peoples of Papua New Guinea
Rule-Based	They are culturally **similar** to the coastal peoples of Papua New Guinea
Input	It is situated at the coast of the Baltic Sea, where it encloses the city of Stralsund
Reference	It is **located** at the coast of the Baltic Sea where it **surrounds** the city of Stralsund
DRESS	It is situated at the coast of the Baltic Sea
Rule-Based	It is **located** at the coast of the Baltic Sea. It **contains** the city of Stralsund

Input	AKI secondary to heart failure medication
Reference	**Kidney injury from related** heart failure medication
Output	**Kidney damage because of** heart failure medication
Input	82F from LLC with worsening SOB and lethargy
Reference	**82 female were admitted to hospital** from **low-level care facility** with worsening **short of breath** and **tiredness**
Output	**82 female** from **low-level care facility** with worsening **breathlessness** and **tiredness**

Table 2: System output comparison from WikiLarge (upper), examples of medical reports simplifications (lower)

clause (where it encloses) in example 2 were rightfully split into separate sentences. On the other hand, the neural system seems to favor deletion operations, even when it affects the meaning. In all three examples, a chunk of the sentence was deleted despite changing the meaning.

As for the medical simplification results, our proposed system has achieved an improvement of 11 points on SARI simplicity scale and 5 grade levels on FKGL scale, when compared to the original input sentences. (two example simplifications are shown in table 2, lower). Looking at the average words per sentence, it is evident that our system tends to produce longer simplified sentences. Once again, that is due to the nature of the medical simplification task which requires elaboration rather than deletion.

7 Conclusion

We developed a non-neural approach for text simplification which implements rules for lexical and syntactic simplification. We used two common test sets to compare our system output with that of a recently proposed neural simplification approach. We showed that our system produces simpler and more meaningful output and scores higher in terms of simplicity metrics. We presented a comparison between both systems output to capture where the neural approach fails. Finally, we presented a hybrid method to enable our system to perform domain-specific text simplification, with high performance, in low-resource situations.

Acknowledgements

The authors are grateful to the reviewers for their insightful comments and feedback. This work is partly supported by the ARC Future Fellowship FT190100039 to G. H. and an MRFF TRIP Fellowship to MAR.

References

Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., & Lai, J. C. (1992). Class-based N-gram Models of Natural Language. *Comput. Linguist.*, *18*(4), 467–479.

Chen, D., & Manning, C. (2014). A Fast and Accurate Dependency Parser using Neural Networks. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 740–750.

Kincaid, J., Fishburne, R., Rogers, R., & Chissom, B. (1975). Derivation Of New Readability Formulas (Automated Readability Index, Fog Count And Flesch Reading Ease Formula) For Navy Enlisted Personnel. *Institute for Simulation and Training*.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., … Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, 177–180.

Kurohashi, S., & Sakai, Y. (1999). Semantic Analysis of Japanese Noun Phrases: A New Approach to Dictionary-based Understanding. *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, 481–488.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: A Method for Automatic Evaluation of Machine Translation. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 311–318.

Scarton, C., Palmero Aprosio, A., Tonelli, S., Martín Wanton, T., & Specia, L. (2017). MUSST: A Multilingual Syntactic Simplification Tool. *Proceedings of the IJCNLP 2017, System Demonstrations*, 25–28.

Siddharthan, A. (2002). An architecture for a text simplification system. *Language Engineering Conference, 2002.* 64–71.

Siddharthan, Advaith. (2014). *A survey of research on text simplification.*

Sulem, E., Abend, O., & Rappoport, A. (2018). *BLEU is Not Suitable for the Evaluation of Text Simplification*.

Vickrey, D., & Koller, D. (2008). Sentence Simplification for Semantic Role Labeling. *Proceedings of ACL-08: HLT*, 344–352.

Xu, W., Callison-Burch, C., & Napoles, C. (2015). Problems in Current Text Simplification Research: New Data Can Help. *Transactions of the Association for Computational Linguistics*, *3*, 283–297.

Xu, W., Napoles, C., Pavlick, E., Chen, Q., & Callison-Burch, C. (2016). Optimizing Statistical Machine Translation for Text Simplification. *Transactions of the Association for Computational Linguistics*, *4*(0), 401–415.

Zhang, X., & Lapata, M. (2017). Sentence Simplification with Deep Reinforcement Learning. *ArXiv:1703.10931 [Cs]*.

Zhu, Z., Bernhard, D., & Gurevych, I. (2010). A Monolingual Tree-based Translation Model for Sentence Simplification. *Proceedings of the 23rd International Conference on Computational Linguistics*, 1353–1361.