Towards Arabic Sentence Simplification via Classification and Generative Approaches

Nouran Khallaf
University of Leeds, UK
Alexandria University, Egypt
mlnak@leeds.ac.uk

Serge Sharoff , Rasha Soliman
University of Leeds, UK
s.sharoff,r.k.soliman@leeds.ac.uk

Abstract

This paper presents an attempt to build a Modern Standard Arabic (MSA) sentence-level simplification system. We experimented with sentence simplification using two approaches: (i) a classification approach leading to lexical simplification pipelines which use Arabic-BERT, a pre-trained contextualised model, as well as a model of fastText word embeddings; and (ii) a generative approach, a Seq2Seq technique by applying a multilingual Text-to-Text Transfer Transformer mT5. We developed our training corpus by aligning the original and simplified sentences from the internationally acclaimed Arabic novel "Saaq al-Bambuu". We evaluate effectiveness of these methods by comparing the generated simple sentences to the target simple sentences using the BERTScore evaluation metric. The simple sentences produced by the mT5 model achieve P 0.72, R 0.68 and F-1 0.70 via BERTScore, while, combining Arabic-BERT and fastText achieves P 0.97, R 0.97 and F-1 0.97. In addition, we report a manual error analysis for these experiments.

1 Introduction

Text Simplification (TS) is a Natural Language Processing (NLP) task that aims to reduce the linguistic complexity of the text while maintaining its meaning and original information (Saggion, 2017; Siddharthan, 2002; Collados, 2013). According to Shardlow (2014) definition, TS involves text transformation with new lexical items and/or rewriting sentences to ensure both its readability and understandability for the target audience (Bott and Saggion, 2011). TS could be classified as a type of Text Style Transfer (TST), where the target style of the generated text is "simple" (Jin et al., 2020). Evidence suggests the importance of TS involves : (i) its usage in designing and simplifying the language curriculum for both second language and first language learners, in making text easy-to-read for first language early learners; in assisting first-

language users with cognitive impairments and low literacy language level; (ii) being a fundamental pre-process in NLP applications such as text retrieval, extraction, summarization, categorization and translation (Saggion, 2017); and (iii) acting as a post-process step in Automatic speech recognition. Hence, there are various types of simplification systems based on the purpose and who is the end-user of the system. There are three key aspects of simple text that: (i) it is made up of frequent simple words, grammatically simple sentences, and direct language; (ii) unnecessary information is omitted ; (iii) it can be shorter by the number of words, but also with shorter sentences, which might lead to their increased number (Bott and Saggion, 2011; Collados, 2013). Collados (2013) approached TS differently as he came up with different opinion, that is a slightly simplified text for one user is generally simpler for any other users. But a more extensive simplification for a specific user, may lead to a more complex text for another user. Most of TS techniques were borrowed from closely related NLP tasks such as Machine Translation (Sikka and Mago, 2020) . This has influenced our experiments to demonstrate the effectiveness of two different methods to address the sentence simplification (SS) task as follows:

(1) Classification Approach SS is considered as a classification task that requires a decision on which word to replace or syntactic structure to regenerate in each complex sentence. This approach allows the application of the Lexical Simplification (LS) task pipeline, i.e that aims to control the readability attribute of the text and make it more accessible to different readers with various intellectual abilities. LS particularly involves word change, thus we experiment the effect of different embedding representation on word classification decision. This approach highlights the impact of how the text is simplified either by applying word embedding, or

contextualised embedding such as BERT (Devlin et al., 2018).

(2) Generative Approach SS is considered as a translation task, in which the translation is done within the same language from a complex sentence as the source to a simplified sentence as the target (Zhu et al., 2010). According to this perspective, SS generative model could be implemented using Machine Translation (MT) and monolingual text-to-text generation techniques. Thus, we combined all SS steps into one process which learns from the complex sentence how to generate the simple version. For this purpose, we applied a BERT-like pre-trained transformer to perform a sequence-to-sequence (Seq2Seq) algorithm.

The main contribution of this paper is to examine different approaches for Arabic sentence simplification task using automatic and manual evaluation. To our knowledge, this is the first available Arabic sentence-level simplification system.

2 Corpus and Tools

The corpus used for training is a set of complex/simple parallel sentences that have been compiled from the internationally acclaimed Arabic novel "Saaq al-Bambuu" which has an authorized simplified version for students of Arabic as a second language (Familiar and Assaf, 2016). We assume that a successful sentence simplifier should be able to detect word/sentences in the original text that require simplification and simplify them in such a way as the original simple counterpart. The dataset consists of 2980 parallel sentences as illustrated in Table 1 and classified according to The Common European Framework of language proficiency Reference (CEFR) .i.e is an international standard for describing language ability ranging from A1, A2 ... up to C2.

Levels	Sentence	Tokens
Simple A+B	2980	34447
Complex C	2980	46521
Total	**5690**	80968

Table 1: Number of Sentences and Tokens available per each CEFR Level in Saaq al-Bambuu parallel corpus

We aligned the words in the parallel "Saaq al-Bambuu" sentences using Eflomal word aligning tool that uses a Bayesian model with Markov Chain Monte Carlo (MCMC) inference (Östling

and Tiedemann, 2016). After aligning the words, we automatically identified four basic simplification types on word-level and sentence-level (Alva-Manchego et al., 2017), then annotate these types with the following labels :

- Deletions, DELETE (D) in the complex sentence. [word-level]
- Additions, ADD (A) in the simplified sentence. [sentence-level]
- Substitutions, REPLACE (R), a word in the complex sentence is replaced by a new word in the simplified sentence. [word-level]
- Rewrites, REWRITE (RW) words shared in both complex and simple sentence pairs. [sentence-level]

The overall calculation of the simplification processes in the "Saaq al-Bambuu" corpus illustrated in figure 1. The $REWRITE$ operation has the highest proportion of the simplification processes [keeping the word as it is in both versions] in which 21899 words were copied in the simplified version. Whereas, 12561 words have been deleted to simplify the sentence that annotated with $DELETE$ label. In the third position comes $REPLACE$ operation in which 9082 words where subsisted with their simple counterparts. At last, only 362 words were added to simplify the sentences that annotated with ADD label.
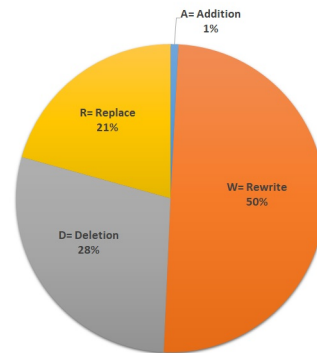

Figure 1: Represents the percentage of each simplification operation on Saaq al-bambuu corpus

Regarding Part-Of-Speech features (POS-features) extraction we used MADAMIRA a robust Arabic morphological analyser and part of speech tagger (Pasha et al., 2014).

3 Method One - Classification approach

The reference for this approach is the pipeline of the LS task, that focuses on LS by replacing com-

plex vocabularies or phrasal-chunks with suitable substances (Paetzold and Specia, 2017b). To reach this goal, we decided to implement three classification models:

1. classification model which is based on *word embedding*, thus we applied ***fastText*** word embedding tool that represents words as vectors embedding. Those vectors embedding was trained on Common Crawl and Wikipedia. We used the Arabic ar.300.bin file in which each word in WE is represented by the 1D vector mapped of 300 attributes(Grave et al., 2018);

2. classification model which is based on *transformers*. Using ***Arabic-BERT*** a pre-trained transformer model on both filtered Arabic Common Crawl and a recent dump of Arabic Wikipedia contain approximately 8.2 Billion words (Safaya et al., 2020) ;

3. classification model combining both *fastText* and *Arabic-BERT* results with post-editing rules;

Considering the definition of the four main steps applied in the pipeline for LS as follows:

Complex word identification [CWI] is the main first step performed at the top of the pipeline that employed to distinguish complex words from simple words in the sentence. *Substitution Generation [SG]* involves generating all possible substitutions but without including ambiguous substances that would confuse the system in the Substitution Selection step. *Substitution Ranking [SR]* is to order the new generated substitution list to ease the selection step by giving high probability of the most appropriate highly ranked word. *Substitution Selection [SS]* is responsible for selecting from the ordered SG's generated list the most appropriate substitute according to the context while preserving the same meaning and grammatical structure.Taking into account the fact that, a word may have multiple meanings, and different meanings will have different relevant substitutions, then the SS task may generate a miss-substitution, which may lead to meaning corruption. The following part of this paper moves on to describe in greater detail the implementation of each step concentrating on employed methods and tools.

3.1 Complex word identification

CWI step could be viewed as a layered analysis opt for a better understanding of word complex-

ity. Hence, we applied a lexicon-based approach. Taking into account one sentence per time, the first level relates to identifying POS-tags along with other features produced by MADAMIRA to be used in further steps. The second layer of analysis moved to assign each word a CEFR complexity level adopting a Lexical based approach using CEFR vocabulary Listas a reference to allocate each word in the target sentence to a readability level. At CWI, with identifying the complex words, these words become the targets to simplify. It is impractical to simplify all complex words in a sentence at once. So that ordering words according to their CEFR level and taking into account each of these words as the target per time to deploy the simplification process. For example, if a sentence has three complex words assigned with B2, C2, C1, firstly we order them to be C2, C1, and B2 and then start the simplification process with targeting C2 tagged word, followed by C1 and so on. In this example, this operation results in generating three sentences each with different masked word slot.

3.2 Substitution Generation and Ranking

These two steps were considered in one process using different methodologies to generate the substitution list and ranking them considering semantic similarity measures. For this purpose we obtained different sentence embedding to produce ten top ranked substitution list of the masked token.

3.2.1 Arabic-BERT prediction

Arabic-BERT model has different tasks to use in various NLP tasks. Here, for each *complex word* use applying BERT's task $MaskedLanguageModeling$ (MLM). This task predicts a substitution list of a masked [not shown, complex] token in a sequence given its left and right context. At this process, the MLM requires a concatenation between the original sequence and the same sentence sequence where the target word is replaced by [MASK] token as a sentence pair, and feed the sentence pair into the BERT to obtain the probability distribution of the possible replacements corresponding to the MASK word. **For example**, given this sentence from Arabic Wikipedia:

تَتَطَلَب مِن هَيئَة المَحكَمَة وُجوُب تَحدِيد الحُقُوق

taṭaṭalabu min hay'atu almaḥkamatu wujūba taḥdīda alḥuqūq

[require the judge or the court to necessarily determine the rights]

The ranking probability of Arabic-BERT's prediction list using fastText was shown on the right side of figure 2.

3.2.2 fastText prediction

Using *fastText* model in two folded processes, first ranking the previously produced substitutions obtained by MLM BERT. This is done by calculating the semantic cosine similarity between each word in the produced list to the target complex word. The second process is using fastText word embedding itself to generate a list of possible replacements [SG] and then ranking by the nearest neighbour [SR]. For example, the fastText generated list given the target complex word in the previous example is shown on the left side of the figure 2.
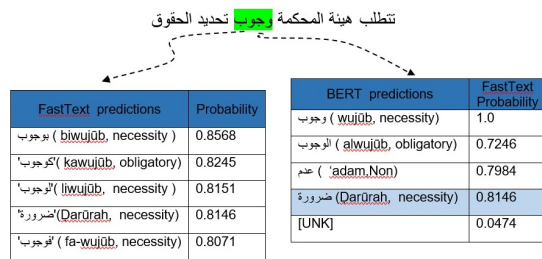
تتطلب هيئة المحكمة وجوب تحديد الحقوق

FastText predictions	Probability
(biwujūb, necessity) بوجوب	0.8568
(kawujūb, obligatory) اكوجوب	0.8245
(liwujūb, necessity) الوجوب	0.8151
(Darūrah, necessity) ضرورة	0.8146
(fa-wujūb, necessity) الوجوب	0.8071

BERT predictions	FastText Probability
(wujūb, necessity) وجوب	1.0
(alwujūb, obligatory) الوجوب	0.7246
('adam,Non) عدم	0.7984
(Darūrah, necessity) ضرورة	0.8146
[UNK]	0.0474

Figure 2: Arabic-BERT and fastText predication lists along with the probability obtained from fastText for the word "wujūba" ('وُجوُب', 'necessity')

3.3 Substitution Selection

At this stage, each complex word in the sentence has different ordered substituted lists based on Arabic-BERT and fastText. Taking into account each prediction list to analyse individually and select the more logical substitute based on the probabilities and some linguistics rules. This allowed the system to generate a set of simplified versions of the target sentence. In addition, keeping a record of the semantic similarity and the readability level of the new produced sentences. The system produces three different simple sentences based on Arabic-BERT substitute selection, fastText, and Combined decision from both generated lists. The combined decision is a very crucial stage and the system needs to be careful when selecting the best substitute based on different measures. Starting with the Arabic-BERT list, the greater the value the most common or familiar is the word for a person referring to simple words. If the word is tagged with replacement with [UNK] the decision is to ignore the results from Arabic-BERT and rely on

fastText results. Then, applying the following four rules to limit incorrect selection:

1. Rule1: if [UNK] is a top-ranked substitute then go to fastText results.Check if the first substitute is [UNK] in this case the system completely ignores BERT results and keep the original then rely on FastText results immediately.

2. Rule 2: if any word's lemma in the generated list equal the lemma of the original word excludes these words from the list.Check if the lemmas in the predicted list matches the same lemma of the target word. In this case, we exclude these words from the potential replacement for the target word and keep only the words with a different lemma. These replacements should also share the same POS and Number with the target word.

3. Rule 3:CEFR list placement for difficulty. Check the word CEFR level of the new substitute word. The new word's CEFR level should be equal to or less than the CEFR level of the target word. Because sometimes the generated list may have a more frequent substitute which is more difficult than the original word but more frequent.

4. Rule 4: check if the new substitute shares the meaning. The system use this rule as it gives a level of confidence to the system selection. After the system makes the final decision either, keep the target word or select the suggested substitute based on previous rules. At this stage, comparing both target and substitute MADAMIRA English translation feature [appeared in Gloss feature]. If both words share part or all possible translation this gives the system confidence to replace the target with the substance.

4 Method Two: Generative Approach

Here, we employ a Seq2Seq approach adopting *T5 "Text-to-Text Transfer Transformer"*. T5 is a BERT-like transformer that takes input a text and training it on the model to generate target text of a different variety of NLP text-based tasks such as (summarization, translation, question answering and more) (Raffel et al., 2019). The main difference between BERT and T5 is that BERT uses a Masked Language Model (MLM) and an encoder-decoder,

although T5 employs a unified Seq2Seq framework (Farahani et al., 2021). T5 model initially targeted English-Language NLP tasks. Recent research extended the model to include more than 101 languages including the Arabic Language. A "multilingual Text-to-Text Transfer Transformer", Multilingual T5, mT5 (Xue et al., 2020), a new variant of T5 and pre-trained on Common Crawl-based dataset. The pre-trained language model was very successful for the Natural Language Understanding (NLU) task.

Considering the multilingual capabilities of mT5 and the suitability of the Seq2Seq format for language generation. This gives it the flexibility to perform any NLP task without having to modify the model architecture in any way. This experiment employs the 'MT5-For-Conditional-Generation' class that is used for language generation. Training a TS model using "Saaq al-Bambuu" parallel sentences, over the mT5-base model. The system was developed in $Python3.8$ environment with using other toolkits such as Natural Language Processing Toolkit $NLTK$ and $Scikit-learn$. Our sentence corpus was randomly split into 80% for training and 20% for testing.

5   Evaluation

Likewise, most TS evaluation approaches have been driven from other similar NLP research areas. Various evaluation methods have been applied across researches to measure the three main aspects of the newly generated text. These aspects are, i) fluency, referring to the grammatically well-formedness and structure simplicity; ii) adequacy, meaning preservation; iii) simplicity, more readable. All methods were evaluated on the same test dataset that consisted of 299 randomly chosen sentences excluded from training. We employed both automatic and manual evaluation comparing both systems.

5.1   Automatic Evaluation

BERTScore is an evaluation metric that computes cosine similarity scores using BERT-style embedding from a pre-trained transformer model. As such models provide a better representation of the linguistic structure, BERTScore evaluation correlates better with human judgments regarding the measurements of sentence similarity. BERTScore evaluation metric overcome the limitations of the previous Machine translation evaluation metrics such as BLEU(Papineni et al., 2002) and SARI(Xu et al., 2016), n-gram based evaluation metrics. These methods were not able to capture two main simplification features: 1) changing word order as paraphrasing simplification method, 2) maintaining the deep structure meaning, despite changes in the surface form structure. Moreover, the BERTScore evaluation method gives the option to use different pre-trained transformer models by applying *baseline rescaling* to adjust the output scores. This allowed determining the performance of different Arabic-language trained BERT models;(i) the default in multilingual BERT (mBERT)(Devlin et al., 2018) that is based on the selected language which is Arabic in this case; (ii) ARBERT, that trained on a collection of six Arabic datasets comprising 61GB of text (6.2B tokens) (Abdul-Mageed et al., 2021); (iii)AraBERTv0.2-base model consist of 77GB of sentences (8.6B tokens) (Antoun et al., 2020). However, AraBERT has been trained on a larger corpus than ARBERT, the latter uses WordPeice tokeniser as illustrated before. Whereas, AraBERT relies on SentencePiece tokeniser that uses spaces as word boundaries. Considering these two parameters reflected in BERTScore metrics.

Classification approach - Automatic Evaluation
The classification system produced three simple versions of the target sentence using BERT-alone, fastText-alone, and combined version. This automatic evaluation was applied to compare different BERT models resolutions of these sentences as represented in Table 2. Figure 3 represents the number of changes performed by each classification model. These primarily results suggests that using fastText-alone perform unneeded simplification resulting in lower F-1. Whereas, a higher F-1 measure in Arabic-BERT-alone generated sentence suggest that using BERT eliminate necessary changes. While the combination of both tools suggestions enhances the substitution ranking and choice process. That eliminates unnecessary changes and enhance performance. In this case, combined produced sentences achieved P 0.97, R 0.97 and F-1 0.97 using ARBERT.

Generative Approach-Automatic Evaluation
Testing the 299 sentences for evaluating the generated simplified sequences compared to the original sentences and the target simple sentences. Using three measures as presented in Table 2.

Classification	P	R	F1
	Default mBert		
Target/fastText	0.962	0.966	0.964
Target /BERT	0.991	0.990	**0.990**
Target / Combined	0.974	0.975	0.975
	ARBERT		
Target/fastText	0.958	0.960	0.959
Target /BERT	0.990	0.991	**0.990**
Target / Combined	0.976	0.976	0.978
	AraBERT		
Target/fastText	0.962	0.963	0.963
Target /BERT	0.989	0.989	0.989
Target / Combined	0.975	0.976	0.976

Generation	P	R	F1
	Default mBert		
Original/Target	0.889	0.838	0.862
Generated/Original	0.806	0.725	0.762
Generated/ Target	0.754	0.723	0.736
	ARBERT		
Original/Target	0.840	0.754	0.790
Generated/Original	0.647	0.529	0.573
Generated/ Target	0.570	0.524	0.538
	AraBERT		
Original/Target	0.879	0.823	0.848
Generated/Original	0.787	0.693	0.734
Generated/ Target	0.723	0.686	0.701

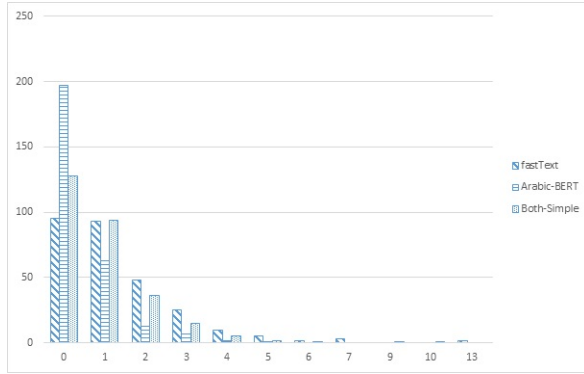Table 2: Precision, recall and F1 measures using BERTScore with different transformer models

Figure 3: number of changed words using fastText-alone, Arabic-Bert-alone and combined

- Original/Target, considering it as a reference to the mT5 system.
- Generated/Original, comparing the newly generated sentence with the original complex sentence.
- Generated/Target, comparing the newly generated sentence with the target simple sentence.

To further illustrate these three models' performance, figure 4, represents the distribution of F-1 across the testing data instances using different BERT models. The default model F-1 plots skewed towards the right reflecting strong similarity across the three parallel sentences (Original/Target/Generated). Whereas, AraBERT plots Original/Target and Generated/Original skewed to the left indicating less similarity across the data. While, ARBERT's plots represent a normal distribution representing a more accurate similarity measure in the data. This findings suggests ARBERT that applying a WordPeice sentence to-

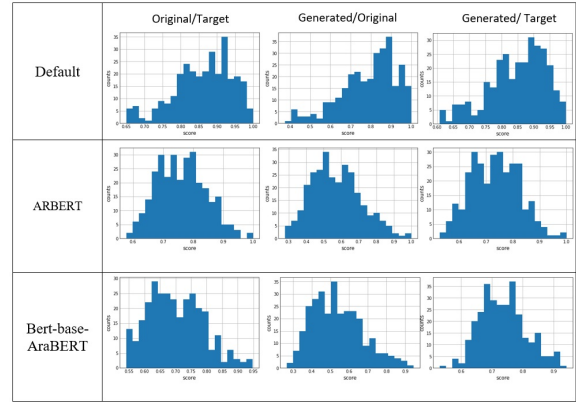keniser BERT model performed better in sentence representation.

Figure 4: The F1 scores for each sentence pair, the scores are more spread out, which makes it easy to compare different methods

5.2 Manual Evaluation

Classification Approach - Manual Evaluation a manual analysis of the produced sentences of combined system has been performed. The results displayed in figure5 on a scale of good, useful, a bit useful, and useless simplification. 55% of the new simplified sentences were either good, useful or a bit useful as a majority. While 45% of the sentences were classified as useless simplification where the complex word was replaced either by a more complex word or its antonym. For example, a useful simplification from the combined system as in this sentence from "Saaq al-Bambuu",

كُنتُ أَحدِق في الطَبَق وَالصَمت يَكَادُ يَبتَلِع المَكَان

Kuntu 'uḥaddiqu fī alṭabaqi wa-al-ṣamtu yakādu yabtali' al-makān.

[I was staring at the plate and the silence almost swallowed up the place.]

In this sentence, the word 'أَحَدق' (*uḥaddiqu,* 'staring') was replaced by (أَتَأَمَل 'ata'mmalu, 'muse'), that is more frequent and simpler and generate:

كُنتُ أَتَأَمَلُ فِي الطَبَق وَالصَمت يَكَادُ يَبتَلِع المَكَان

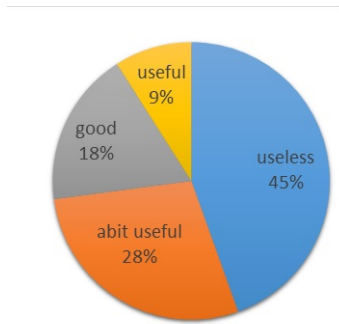Although, it is simpler it doesn't reach the exact target word أَنظُر ('Anẓuru, 'look')

Figure 5: Simplified sentences analysis based on the usefulness of the lexical substitution processes.

Generative Approach-Manual Evaluation despite the initial automatic evaluation provided promising results, the manual evaluation of the generated text provides deeper insight into mT5's output for the Arabic simplification task. According to the manual error analysis as shown in figure 6 only 31 sentences were correctly simplified from 299 testing instances. In addition, about 120 generated sentences were incomplete and the system produced 64 meaningless or ill-formed sentences. A significant shortcoming that the produced sentences tends to have the same repeated phrase. Moreover, one of the generated sentences were more complex than the original sentence.

Otherwise, mT5 in some cases can produce a perfectly valid paraphrase, which is better than the target simple sentence.

طَلَب مِنَا الجلُوس فِي صَالُونَه المَلِيء بِالكُتُب

ṭalab minnā al-julūs fī ṣālwnahu almaly' bi-al-kutub

[He asked us to sit in his salon full of books.]

فِي صَالُونَه الصَغِير المَلِيء بِالكُتُب طَلَب مِنَا الجلُوس أَمَام مَكتَب صَغِير

Fī ṣālunahu al-ṣaghīr almali' bil-kutubi, ṭalaba minnā al-julūsi 'amāma maktabi ṣaghīri

[In his small salon full of books, he asked us to sit in front of a small desk.] In this case, the generated sentence was syntactically simpler than the target while focusing on the main information.
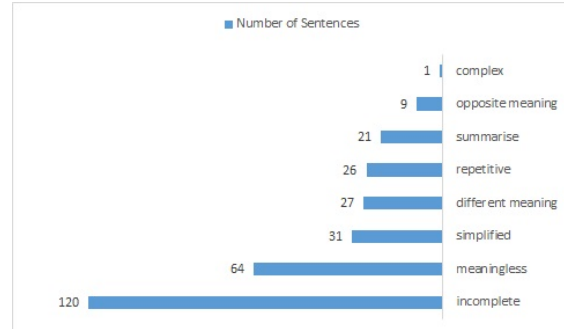

Figure 6: Manual error analysis distribution across testing data

6 Related Works

Blum and Levenston (1978) completed one of the first studies that introduce Lexical simplification for Teaching English as a Second Language (TESOL). Some of the following TS systems applied a rule-based approach (Petersen and Ostendorf, 2007; Evans et al., 2014). Most later carried out studies based on a monolingual parallel-aligned corpus of original and simplified texts by applying different machine-learning algorithms such as Aluísio et al. (2008) and Caseli et al. (2009) for Portuguese language, Collados (2013) for Spanish language and Glavaš and Štajner (2015) for English. Other researchers considered the TS problem as a monolingual translation problem that is best solved through applying the Statistical Machine Translation (SMT) framework (Specia, 2010; Zhu et al., 2010; Woodsend and Lapata, 2011; Wubben et al., 2012). Latest English TS studies start applying word embedding(Paetzold and Specia, 2016, 2017a) and BERT transformers for lexical simplification as presented in Qiang et al. (2020) proving its effectiveness in solving LS task.

Unlike English and Other Latin languages, only a few researchers have been tackling the problems of Arabic ATS. Al-Subaihin and Al-Khalifa (2011) a prototype unreleased system at King Saud University, they proposed Arabic Automatic Text simplification system (AATS) called Al-basset. The system architecture for AATS structured in the light of the state of the art of systems for other

languages. Such as SYSTAR, a syntactic simplification system for the English aphasic or inarticulate population(Carroll et al., 1998). Another system, SIMPLIFICA, is a simplification tool for Brazilian Portuguese (BP) targeting those with low literacy levels (Scarton et al., 2010). The design of "Al-Basset" was constructed of four main stages: i) measuring complexity, in this stage they would adopt a statistical language model based on a machine learning technique called ARABILITY (Al-Khalifa and Al-Ajlan, 2010); ii) vocabulary (lexical) simplification by following the LS-pipeline and produce the synonyms either by building a new dictionary or using Arabic-WordNet(Rodríguez et al., 2008) while select the most common and possible synonym, by using the Google API; iii) syntactic simplification, they suggested identifying the complex structures by applying a look-up approach to a manually predefined list of Arabic complex structures; iv) diacratization using MADA(Habash et al., 2009) diacritizer task. The main limitation of implementing this system at this point is the unavailability of Arabic basic resources and tools. Such as dictionaries, corpora and parallel complex-simple structures which are the main components of any ATS system.

Al Khalil et al. (2017) provided the second attempt to build an AATS system at New York University in Abu-Dhabi. Their simplification system was designed to be semi-automatic to simplify Arabic modern fiction; it involved a linguist using a web-based application to apply ACTFL (American Council on the Teaching of Foreign Languages) language proficiency guidelines for simplification of five Arabic novels. They aimed to provide essential Arabic resources for building ATS and formulating manual simplification rules for Arabic fiction novels using TS stat-of-the-art. The first resource they expected to produce is a corpus consisting of 1M tokens of the 12-grade curriculum, 5M tokens of the adult novels (original and simplified counterparts), and 500K tokens of children's stories. Also, they provided a proposal to the SAMER (Simplification of Arabic Masterpieces for extensive reading) project based on the corpus analysis. Their guidelines invoke both the MADAMIRA (Pasha et al., 2014) and CAMAL dependency parser (Shahrour et al., 2016) for data analysis and classification of their corpus. They were aiming to build a readability measurement identifier to formulate a 4-levelled graded reader scale (GRS) by applying various machine-learning classifiers.

7 Conclusion

In this paper, we have presented the first Modern Standard Arabic sentence simplification system by applying both classification and generative approaches. On the one hand, the classification approach focuses on lexical simplification. We looked at the different classification methods and showed that a combined method generates well-formed simple sentences. In addition, using word embeddings and transformers prove to produce a reasonable set of substitutions for the complex word more accurately than traditional methods such as WordNet. Our interpretation of the limitation of the classification system arises from the fact that some of the generated sentence structures are not well-formed and that the system can misidentify what makes some complex words in the CWI step. Even though this limitation reveals the limitations of the Arabic CEFR vocabulary list in identifying the complex word, the list is shown to be more useful in the substitution replacement step.

On the other hand, while the generative Seq2Seq approach provides a less accurate simplified version in most cases, in some cases it outperforms the classification approaches by generating a simplified sentence, which can be even better than the target human simple sentence. Nevertheless, one of the limitations of the generative approach concerns the trend to repeat identical patterns, which can be partly controlled by post-processing.

8 Limitations

We have discussed the relative limitations of the two approaches in the paper. Overall, our paper relies on a single parallel resource. When other datasets become available, it will be important to experiment with them. With the use of pre-trained models, the requirements for training models from scratch are relatively low.

9 Ethics Statement

This paper is the authors' own original work, which has not been previously published elsewhere.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT &

MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.

Hend S Al-Khalifa and Amani A Al-Ajlan. 2010. Automatic readability measurements of the arabic text: An exploratory study. *Arabian Journal for Science and Engineering*, 35(2 C):103–124.

Muhamed Al Khalil, Nizar Habash, and Hind Saddiki. 2017. Simplification of arabic masterpieces for extensive reading: A project overview. *Procedia Computer Science*, 117:192–198.

Afnan A Al-Subaihin and Hend S Al-Khalifa. 2011. Al-baseet: A proposed simplification authoring tool for the arabic language. In *2011 International Conference on Communications and Information Technology (ICCIT)*, pages 121–125. IEEE.

Sandra M Aluísio, Lucia Specia, Thiago AS Pardo, Erick G Maziero, and Renata PM Fortes. 2008. Towards brazilian portuguese automatic text simplification systems. In *Proceedings of the eighth ACM symposium on Document engineering*, pages 240–248.

Fernando Alva-Manchego, Joachim Bingel, Gustavo Paetzold, Carolina Scarton, and Lucia Specia. 2017. Learning how to simplify from explicit labeling of complex-simplified text pairs. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 295–305.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Shoshana Blum and Eddie A Levenston. 1978. Universals of lexical simplification. *Language learning*, 28(2):399–415.

Stefan Bott and Horacio Saggion. 2011. An unsupervised alignment algorithm for text simplification corpus construction. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 20–26.

John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplification of english newspaper text to assist aphasic readers. In *Proceedings of the AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, pages 7–10. Citeseer.

Helena M Caseli, Tiago F Pereira, Lucia Specia, Thiago AS Pardo, Caroline Gasperin, and Sandra Maria Aluísio. 2009. Building a brazilian portuguese parallel corpus of original and simplified texts. *Advances in Computational Linguistics, Research in Computer Science*, 41:59–70.

José Camacho Collados. 2013. *Syntactic Simplification for Machine Translation*. Ph.D. thesis, Wolverhampton, United Kingdom.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Richard Evans, Constantin Orasan, and Iustin Dornescu. 2014. An evaluation of syntactic simplification rules for people with autism. In *Online Open-source*. Association for Computational Linguistics.

Laila Familiar and Tanit Assaf. 2016. *Saud al-Sanousi's Saaq al-Bambuu: The Authorized Abridged Edition for Students of Arabic*. Georgetown University Press.

Mehrdad Farahani, Mohammad Gharachorloo, Marzieh Farahani, and Mohammad Manthouri. 2021. Parsbert: Transformer-based model for persian language understanding. *Neural Processing Letters*, 53(6):3831–3847.

Goran Glavaš and Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 63–68.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. *arXiv preprint arXiv:1802.06893*.

Nizar Habash, Owen Rambow, and Ryan Roth. 2009. Mada+ tokan: A toolkit for arabic tokenization, diacritization, morphological disambiguation, pos tagging, stemming and lemmatization. In *Proceedings of the 2nd international conference on Arabic language resources and tools (MEDAR), Cairo, Egypt*, volume 41, page 62.

Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2020. Deep learning for text style transfer: A survey. *arXiv preprint arXiv:2011.00416*.

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with Markov Chain Monte Carlo. *Prague Bulletin of Mathematical Linguistics*, 106:125–146.

Gustavo Paetzold and Lucia Specia. 2016. Semeval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569.

Gustavo Paetzold and Lucia Specia. 2017a. Lexical simplification with neural ranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 34–40.

Gustavo H Paetzold and Lucia Specia. 2017b. A survey on lexical simplification. *Journal of Artificial Intelligence Research*, 60:549–593.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Arfath Pasha, Mohamed Al-Badrashiny, Mona T Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In *Lrec*, volume 14, pages 1094–1101.

Sarah E Petersen and Mari Ostendorf. 2007. Text simplification for language learners: a corpus analysis. In *Workshop on Speech and Language Technology in Education*. Citeseer.

Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu. 2020. Lexical simplification with pretrained encoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8649–8656.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Horacio Rodríguez, David Farwell, Javi Farreres, Manuel Bertran, Musa Alkhalifa, M Antonia Martí, William Black, Sabri Elkateb, James Kirk, Adam Pease, et al. 2008. Arabic wordnet: Current state and future extensions. In *Proceedings of The Fourth Global WordNet Conference, Szeged, Hungary*.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. Kuisail at semeval-2020 task 12: Bert-cnn for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059.

Horacio Saggion. 2017. Automatic text simplification. *Synthesis Lectures on Human Language Technologies*, 10(1):1–137.

Carolina Scarton, Matheus Oliveira, Arnaldo Candido Jr, Caroline Gasperin, and Sandra Aluísio. 2010. Simplifica: a tool for authoring simplified texts in brazilian portuguese guided by readability assessments. In *Proceedings of the NAACL HLT 2010 Demonstration Session*, pages 41–44.

Anas Shahrour, Salam Khalifa, Dima Taji, and Nizar Habash. 2016. Camelparser: A system for arabic syntactic analysis and morphological disambiguation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 228–232.

Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 4(1):58–70.

Advaith Siddharthan. 2002. An architecture for a text simplification system. In *Language Engineering Conference, 2002. Proceedings*, pages 64–71. IEEE.

Punardeep Sikka and Vijay Mago. 2020. A survey on text simplification. *arXiv preprint arXiv:2008.08612*.

Lucia Specia. 2010. Translating from complex to simplified sentences. In *International Conference on Computational Processing of the Portuguese Language*, pages 30–39. Springer.

Kristian Woodsend and Mirella Lapata. 2011. Wikisimple: Automatic simplification of wikipedia articles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25.

Sander Wubben, EJ Krahmer, and APJ van den Bosch. 2012. Sentence simplification by monolingual machine translation. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1353–1361.