Benchmarking Automated Clinical Language Simplification: Dataset, Algorithm, and Evaluation

Junyu Luo[1], **Junxian Lin**[2,3], **Chi Lin**[2,3], **Cao Xiao**[4], **Xinning Gui**[1], **Fenglong Ma**[1*]
[1]College of Information Sciences and Technology, Pennsylvania State University, USA
[2]School of Software Technology, Dalian University of Technology, China
[3]Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, China
[4]Relativity, USA
{junyu, xinninggui, fenglong}@psu.edu, linjunxian@mail.dlut.edu.cn,
c.lin@dlut.edu.cn, cao.xiao@relativity.com

Appendix

1 Sentence Selection

The sentence selection algorithm is summarized in Algorithm 1. For each tokenized word w_i, we first check whether it belongs to the medical abbreviation set A, and abb in line 4 denotes the current number of medical abbreviations in the target sentence s. If w_i is not a medical abbreviation, then we will check whether the lemmatized w_i belongs to the top-3000 word set T. unc in line 8 represents the current number of words that are neither medical abbreviations nor commonly-used words. Finally, based on the predefined criteria, the algorithm can automatically decide to keep or remove the target sentence.

Algorithm 1: Sentence Selection Algorithm

```
   Input: Target sentence $s$, top-3000 word set
          $T$, medical abbreviation set $A$
   Output: Selected sentence set
 1 Tokenize $s$ into words $[w_1, ..., w_n]$;
 2 for $i = 1$ to $n$ do
 3     if $w_i \in A$ then
 4         $abb$ = $abb$ + 1;
 5     end
 6     $w_i' = lemmatize(w_i)$;
 7     if $w_i' \notin T$ then
 8         $unc = unc + 1$;
 9     end
10 end
11 if $n < 10$ or $\frac{unc+abb}{n} > 0.5$ or
     $\frac{unc+abb}{n} < 0.1$ then
12     return False;
13 else
14     return True;
15 end
```

2 Restricted Translating

The restricted translating algorithm is summarized in Algorithm 2. In line 5, we first check whether the algorithm meets the special token to decide entering the copy or translating state. Lines 6-8 illustrate the process of copy state, i.e., copying the original input and updating the state of the translator. Lines 10-15 illustrate the translating state, and the translator model will use its own output to update the state and copy it to the answer Y.

Algorithm 2: Partial Translation

```
   Input: Input Sentence $\tilde{W}$, Translation
          Model $Tran$
   Output: Input Sentence $Y$
 1 $W$ = Tokenizer($\tilde{W}$);
 2 $i = 0$;
 3 $Y$ = [];
 4 while Not complete generating $Y$ do
 5     if In Copy Stage then
 6         $Y$.append($W[i]$);
 7         $Tran$.next_state($W[i]$);
 8         $i++$;
 9     else
10         while Not Finish Translating do
11             $y$ = $Tran$.next_word();
12             $Y$.append($y$);
13             $Tran$.next_state($y$);
14         end
15         Skip $i$ to the next copy word;
16     end
17 end
18 return $Y$;
```

3 Baselines & Parameter Settings

Baselines. *Dictionary-based model* means randomly select one full-term expression for each located token by the locator and directly replace the

[*] Corresponding author.

terms in the sentence as the output. *Moses* is a widely-used statistical machine translation (SMT) system.

Neural machine translation approaches include *Seq2Seq* (**?**) and its two variants, i.e., *Seq2Seq*$-$ without using the attention mechanism in the decoder and *Seq2Seq-S* that shares the embedding space of encoder and decoder models. *PointerNet* is a modified version of the pointer network (**?**) by adding a generating/referring option to the model. In the referring mode, the model acts as a general pointer network. However, in the generating mode, the model acts like a normal Seq2Seq model. For general ATS methods, we select the EditNTS (**?**), which is based on the sentence modification operations. For transformer-based models, we include *BART* (**?**), which is a pre-trained transformer autoencoder framework and designed for natural language generation tasks; and *T5* (**?**) that is also a transformer autoencoder framework proposed by Google. Compared to *BART*, *T5* contains more advanced pre-trained tasks and has been proved to be a powerful framework on many natural language generations and understanding tasks. The last baseline is *BERT-MT*, which contains a BERT encoder and a LSTM decoder. It is similar to the polisher but directly translates the original inputs to the targets.

Parameter Settings. For the statistical model Moses, we follow the training procedure listed on the User Manual and Code Guide file[1]. For the dictionary method, we use the pre-constructed dictionary as the same as the DECLARE model. For neural machine translation models and text summarization baseline, we all conduct a grid search to find the optimal parameters. For the EditNTS, we use the default original setting with the learning rate of $1e-3$ with Adam optimizer. The dimension setting is as same as the original work, a 200 dimension bi-direction RNN. For the BERT-MT model, the hidden size is the same as that of PubMedBERT, which is 786. We also use the default AdamW optimizer used by PubMedBERT with the learning rate as $5e-5$, the warm-up method, the default PubMedBERT vocabulary, and tokenization are applied. For BART and T5, the setting of the optimizer and training procedure is the same as the BERT-MT.

Finally, for Seq2Seq, Seq2Seq$-$, Seq2Seq-S, and PointerNet, the hidden size is set to 256 for both encoder and decoder by greedy search, and the learning rate is set to $1e-3$. We use Adam (**?**) as the optimizer. Tokenization is performed using NLTK word tokenizer (**?**). The early stop is also applied by checking the BLEU score (**?**) on the validation set, and the training batch size is set to 30.

For the proposed DECLARE, the locator is based on PubMedBERT to perform token level classification, and we use the default setting of PubMedBERT to train the locator. For the dictionary-based neural interpreter, we use the same parameter setting as (**?**). The max size of the answers is set to 8. The maximum length of the input sentence is set to 64 during training. The learning rate is set to $5e-5$ with 10 epochs, and an early stop is adopted. For the restricted polisher, the setting is the same as the BERT-MT model except the restricted translation setting. PubMedBERT is applied with an LSTM decoder that has the same hidden size.

In the evaluation stage, the same NLTK word tokenizer is applied as baselines to break the sentences into words for calculating the scores for a fair comparison. All models are trained on Ubuntu 16.04 with 128GB memory and an Nvidia Tesla P100 GPU.

4 Experimental Results

Insight Analysis To analyze the influence of the sentence length on the model performance, we divide the source sentences into five groups with different length and calculate the average scores among different length groups. The results are shown in Figure 1.

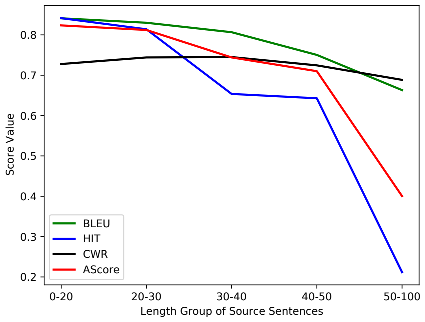

Figure 1: Sentence length v.s. performance.

We can observe that with the increase of the sentence length, the values of BLEU, HIT, and AScore drop, which is in accord with traditional

[1] http://www.statmt.org/moses/manual/manual.pdf

machine translation tasks. However, the trend of the CWR score is different from that of the other three metrics, which keeps a stable performance. The reason is that CWR reflects a language style feature, which is relatively independent from the length. From this experiment, we can conclude that a single CWR score can not reveal the actual performance of the model in our task. These results also confirm the reasonableness of the design of AScore, which assigns more importance weights to the BLEU and HIT scores compared with the CWR score.

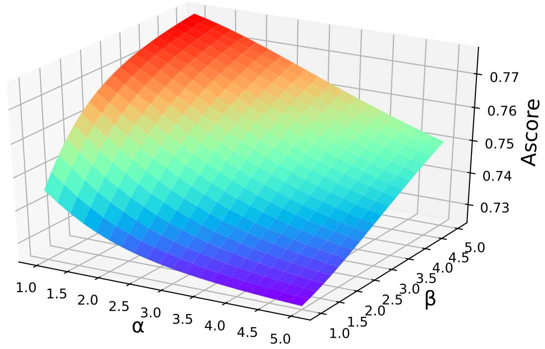

Figure 2: Ascore changes regrading α and β.

Hyperparameter Analysis. In the proposed new metric AScore, there are two key parameters α and β. Figure 2 shows the values of AScore with regard to the changes of α and β. We can observe that given a fixed α, the values of AScore increase with the increase of β. When we fix the value of β, the values of AScore will decrease with the increase of α. The values of AScore are in the range [BLEU, HIT]. Thus, AScore is a trade-off between readability (BLEU), correctness (HIT), and simplicity (CWR).

Case Study To further demonstrate the effectiveness of the proposed DECLARE, we conduct a case study as shown in Table 1. We can observe that DECLARE successfully and correctly translates all the professional terms. The BLEU score of DECLARE is 0.8235, which is higher than those of other baselines.

DECLARE first uses the locator to identify the professional terms “NSTEMI”, “CAD”, “3V-CABG”, and “RCA”, which are hard to be understood by patients with low health literacy. Then the neural interpreter selects the best replacement from the mapping dictionary for each professional clinical term, such as replacing “CAD” with “coronary artery disease”. Finally, the polisher is in charge of simplifying the clinical language to layperson-understandable languages, such as translating “coronary artery disease” to “heart disease”. BERT-MT also generates high-quality sentences, but there are missed professional and redundant words, such as “3v - ” and “right”. Thus, the readability of BERT-MT’s output is lower than that of DECLARE’s.

Table 2 shows a hard example that almost all the approaches fail to translate the source sentence. Compared with other baselines, DECLARE can generate the word “non-alcoholic”, which leads to its performance better than others. However, all the approaches are unfamiliar with the word “cirrhosis”, which is the main reason for the failure. From these results, we can find that it is challenging to accurately translate clinical jargon to layperson-understandable language.

In both cases, the EditNTS failed to simplify any professional words. This result can prove that despite the general ATS methods are good at simplifying complex general words and sentences, they are limited in their simplification of professional medical terminologies.

5 FKGL Score Results

For the FKGL (lower the better), we can find that the ground truth is the worst result as shown in Table 3, which can effectively illustrate why we argue that it is not a suitable metric for our task. The simplification of the professional medical terms involves replacing the short abbreviations into long common words. However this will actually increase the FKGL score since FKGL focus on the average words in a sentence and the average syllables in a word. Transferring the abbreviations into long common words will increase the above metrics and results a worse FKGL score.

Source:	**NSTEMI**/**CAD** - history of **3V-CABG** with only **RCA** graft still patent .
Reference 1:	[non-ST-elevation myocardial infarction]/[coronary artery disease] - history of [coronary artery bypass graft] with only [right coronary artery] graft still patent .
Reference 2:	heart attack/heart disease - history of heart bypass surgery with only right heart artery graft still patent .
DECLARE	heart attack attack/heart disease-history of coronary artery bypass graft with only right heart artery graft still patent .
BERT-MT	heart attack/heart disease - history of 3v - heart bypass surgery with only right right heart artery graft still patent .
EditNTS	nstemi/cad - history of 3v-cabg with only right heart artery still patent . eost
T5	NSTEMI/CAD-history of 3V-CAD with only RCA graft still patent
BART	NSTEMI/coronary artery disease-history of 3V-catheter graft with only right coronary artery graft still patent
Seq2Seq	- history of with only right heart artery graft .
Seq2Seq−	- - history of with history only - when are only .
PointerNet	- history of right heart disease graft with two-vessel coronary artery still patent .
Moses	nstemi/cad - history of 3v-cabg with only still patent artery graft .

Table 1: An example that DECLARE outperforms other baselines.

Source:	# **cirrhosis** : patient with history of alcoholic vs **nash** **cirrhosis** complicated by esophagel , gastric , and rectal varices
Reference 1:	# [chronic disease of the liver] : patient with history of alcoholic vs [non-alcoholic steatohepatitis] [chronic disease of the liver] complicated by esophagel , gastric , and rectal varices .
Reference 2:	# [chronic disease of the liver] : patient with history of alcoholic vs liver inflammation and damage complicated by esophagel , gastric , and rectal varices .
DECLARE	cirrhosis : patient with history of alcoholic vs [non-alcoholic] cirrhosis cirrhosis cirrhosis complicated by esophagel , gastric , and rectal varices .
BERT-MT	cirrhosis cirrhosis : patient with history of alcoholic vs late disease disease complicated by food , , , , , , , esesesesesesesesesesesesses .
EditNTS	# cirrhosis : patient with history of alcoholic vs nash cirrhosis complicated by esophagel , gastric , and rectal varices . eost
T5	#cirrhosis: patient with history of alcoholic vsnash cirrhosis complicated by inflammation of the esophagus, gastric, and rectal varices.
BART	#cirrhosis: patient with history of alcoholic vsnash cirrhosis complicated by esophagel, gastric, and rectal varices
Seq2Seq	# cirrhosis : patient with history of alcoholic vs cirrhosis cirrhosis , gastric , gastric , and , , , varices .
Seq2Seq−	# surgical history of patient with history by , history , patient , by surgical by and by surgical tract .
PointerNet	# cirrhosis : patient with history of painful cell function cirrhosis complicated by , , , , , , and rectal rectal in rectal varices .
Moses	# cirrhosis : patient with history of alcoholic cirrhosis , complicated by nash esophagel , acid , and rectal and .

Table 2: A hard example that all the approaches cannot translate accurately.

Model	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**	**BLEU**	**METEOR**	**ROUGE-L**	**CIDEr**	**SARI**	**HIT**	**CWR**	**AScore**	**FKGL↓**
GroundTruth	-	-	-	-	-	-	-	-	-	-	-	-	10.9603
Dictionary	0.7158	0.6364	0.5684	0.5076	0.6070	0.3933	0.7308	4.2037	37.3391	0.5572	0.6407	0.5948	10.3001
Moses	0.7880	0.7130	0.6530	0.6016	0.6889	0.4237	0.8188	5.1046	51.6827	0.6823	0.7543	0.6859	9.0255
Seq2seq	0.7136	0.6322	0.5969	0.5160	0.6147	0.3533	0.7609	4.1299	46.1328	0.7388	0.7980	0.6648	8.1309
Seq2seq-	0.5066	0.3315	0.2373	0.1787	0.3135	0.1859	0.4948	1.2670	24.5346	0.6427	**0.8367**	0.4070	6.4085
Seq2seq-S	0.7180	0.6386	0.5778	0.5267	0.6153	0.3604	0.7683	4.2635	46.5085	0.7331	0.7953	0.6630	8.8005
PointerNet	0.6870	0.5904	0.5158	0.4541	0.5618	0.3338	0.7285	3.9458	42.2857	0.6414	0.7555	0.5949	9.4993
EditNTS	0.8213	0.7801	0.7452	0.7132	0.7649	0.4674	0.7401	5.9508	62.6036	0.6405	0.6915	0.7116	5.4448
BART	0.7148	0.6755	0.6396	0.6060	0.6590	0.5320	0.7616	4.9783	70.3058	0.5266	0.7311	0.6191	10.5039
T5	0.7223	0.6812	0.6445	0.6103	0.6646	0.5305	0.7645	5.0629	71.3255	0.5262	0.7342	0.6220	10.7484
BERT-MT	0.8003	0.7428	0.6952	0.6531	0.7228	0.4566	0.8218	5.3293	**72.2260**	0.7808	0.7358	0.7417	9.0255
DECLARE	**0.8624**	**0.8291**	**0.8004**	**0.7737**	**0.8165**	**0.5290**	**0.8894**	**6.7212**	70.8583	**0.7986**	0.7328	**0.7983**	10.3187
↑	+7.8%	+11.6%	+15.7%	+18.5%	+12.9%	+15.9%	+8.2%	+26.1%	-1.9%	+2.2%	-12.4%	+7.6%	

Table 3: Results with FKGL score.