Controllable Sentence Simplification

Abstract

Text simplification aims at making a text easier to read and understand by simplifying grammar and structure while keeping the underlying information identical. It is often considered an all-purpose generic task where the same simplification is suitable for all; however multiple audiences can benefit from simplified text in different ways. We adapt a discrete parametrization mechanism that provides explicit control on simplification systems based on Sequence-to-Sequence models. As a result, users can condition the simplifications returned by a model on parameters such as length, amount of paraphrasing, lexical complexity and syntactic complexity. We also show that carefully chosen values of these parameters allow out-of-the-box Sequence-to-Sequence models to outperform their standard counterparts on simplification benchmarks. Our model, which we call ACCESS (as shorthand for AudienCe-CEntric Sentence Simplification), increases the state of the art to 41.87 SARI on the WikiLarge test set, a +1.42 gain over previously reported scores.

Introduction

In Natural Language Processing, the Text Simplification task aims at making a text easier to read and understand. Text simplification can be beneficial for people with cognitive disabilities such as aphasia BIBREF0, dyslexia BIBREF1 and autism BIBREF2 but also for second language learners BIBREF3 and people with low literacy BIBREF4. The type of simplification needed for each of these audiences is different. Some aphasic patients struggle to read sentences with a high cognitive load such as long sentences with intricate syntactic structures, whereas second language learners might not understand texts with rare or specific vocabulary. Yet, research in text simplification has been mostly focused on developing models that generate a single generic simplification for a given source text with no possibility to adapt outputs for the needs of various target populations.

In this paper, we propose a controllable simplification model that provides explicit ways for users to manipulate and update simplified outputs as they see fit. This work only considers the task of Sentence Simplification (SS) where the input of the model is a single source sentence and the output can be composed of one sentence or splitted into multiple. Our work builds upon previous work on controllable text generation BIBREF5, BIBREF6, BIBREF7, BIBREF8 where a Sequence-to-Sequence (Seq2Seq) model is modified to control attributes of the output text. We tailor this mechanism to the task of SS by considering relevant attributes of the output sentence such as the output length, the amount of paraphrasing, lexical complexity, and syntactic complexity. To this end, we condition the model at train time, by feeding those parameters along with the source sentence as additional inputs.

Our contributions are the following: (1) We adapt a parametrization mechanism to the specific task of Sentence Simplification by choosing relevant parameters; (2) We show through a detailed analysis that our model can indeed control the considered attributes, making the simplifications potentially able to fit the needs of various end audiences; (3) With careful calibration, our controllable parametrization improves the performance of out-of-the-box Seq2Seq models leading to a new state-of-the-art score of 41.87 SARI BIBREF9 on the WikiLarge benchmark BIBREF10, a +1.42 gain over previous scores, without requiring any external resource or modified training objective.

Related Work ::: Sentence Simplification

Text simplification has gained more and more interest through the years and has benefited from advances in Natural Language Processing and notably Machine Translation.

In recent years, SS was largely treated as a monolingual variant of machine translation (MT), where simplification operations are learned from complex-simple sentence pairs automatically extracted from English Wikipedia and Simple English Wikipedia BIBREF11, BIBREF12.

Phrase-based and Syntax-based MT was successfully used for SS BIBREF11 and further tailored to the task using deletion models BIBREF13 and candidate reranking BIBREF12. The candidate reranking method by BIBREF12 favors simplifications that are most dissimilar to the source using Levenshtein distance. The authors argue that dissimilarity is a key factor of simplification.

Lately, SS has mostly been tackled using Seq2Seq MT models BIBREF14. Seq2Seq models were either used as-is BIBREF15 or combined with reinforcement learning thanks to a specific simplification reward BIBREF10, augmented with an external simplification database as a dynamic memory BIBREF16 or trained with multi-tasking on entailment and paraphrase generation BIBREF17.

This work builds upon Seq2Seq as well. We prepend additional inputs to the source sentences at train time, in the form of plain text special tokens. Our approach does not require any external data or modified training objective.

Related Work ::: Controllable Text Generation

Conditional training with Seq2Seq models was applied to multiple natural language processing tasks such as summarization BIBREF5, BIBREF6, dialog BIBREF18, sentence compression BIBREF19, BIBREF20 or poetry generation BIBREF21.

Most approaches for controllable text generation are either decoding-based or learning-based.

Decoding-based methods use a standard Seq2Seq training setup but modify the system during decoding to control a given attribute. For instance, the length of summaries was controlled by preventing the decoder from generating the End-Of-Sentence token before reaching the desired length or by only selecting hypotheses of a given length during the beam search BIBREF5. Weighted decoding (i.e. assigning weights to specific words during decoding) was also used with dialog models BIBREF18 or poetry generation models BIBREF21 to control the number of repetitions, alliterations, sentiment or style.

On the other hand, learning-based methods condition the Seq2Seq model on the considered attribute at train time, and can then be used to control the output at inference time. BIBREF5 explored learning-based methods to control the length of summaries, e.g. by feeding a target length vector to the neural network. They concluded that learning-based methods worked better than decoding-based methods and allowed finer control on the length without degrading performances. Length control was likewise used in sentence compression by feeding the network a length countdown scalar BIBREF19 or a length vector BIBREF20.

Our work uses a simpler approach: we concatenate plain text special tokens to the source text. This method only modifies the source data and not the training procedure. Such mechanism was used to control politeness in MT BIBREF22, to control summaries in terms of length, of news source style, or to make the summary more focused on a given named entity BIBREF6. BIBREF7 and BIBREF8 similarly showed that adding special tokens at the beginning of sentences can improve the performance of Seq2Seq models for SS. Plain text special tokens were used to encode attributes such as the target school grade-level (i.e. understanding level) and the type of simplification operation applied between the source and the ground truth simplification (identical, elaboration, one-to-many, many-to-one). Our work goes further by using a more diverse set of parameters that represent specific grammatical attributes of the text simplification process. Moreover, we investigate the influence of those parameter on the generated simplification in a detailed analysis.

Adding Explicit Parameters to Seq2Seq

In this section we present ACCESS, our approach for AudienCe-CEntric Sentence Simplification. We parametrize a Seq2Seq model on a given attribute of the target simplification, e.g. its length, by prepending a special token at the beginning of the source sentence. The special token value is the ratio of this parameter calculated on the target sentence with respect to its value on the source sentence. For example when trying to control the number of characters of a generated simplification, we compute the compression ratio between the number of characters in the source and the number of characters in the target sentence (see Table TABREF4 for an illustration). Ratios are discretized into bins of fixed width of 0.05 in our experiments and capped to a maximum ratio of 2. Special tokens are then included in the vocabulary (40 unique values per parameter).

At inference time, we just set the ratio to a fixed value for all samples. For instance, to get simplifications that are 80% of the source length, we prepend the token $<$NbChars_0.8$>$ to each source sentence. This fixed ratio can be user-defined or automatically set. In our setting, we choose fixed ratios that maximize the SARI on the validation set.

We conditioned our model on four selected parameters, so that they each cover an important aspect of the simplification process: length, paraphrasing, lexical complexity and syntactic complexity.

NbChars: character length ratio between source sentence and target sentence (compression level). This parameter accounts for sentence compression, and content deletion. Previous work showed that simplicity is best correlated with length-based metrics, and especially in terms of number of characters BIBREF23. The number of characters indeed accounts for the lengths of words which is itself correlated to lexical complexity.

LevSim: normalized character-level Levenshtein similarity BIBREF24 between source and target. LevSim quantifies the amount of modification operated on the source sentence (through paraphrasing, adding and deleting content). We use this parameter following previous claims that dissimilarity is a key factor of simplification BIBREF12.

WordRank: as a proxy to lexical complexity, we compute a sentence-level measure, that we call WordRank, by taking the third-quartile of log-ranks (inverse frequency order) of all words in a sentence. We subsequently divide the WordRank of the target by that of the source to get a ratio. Word frequencies have shown to be the best indicators of word complexity in the Semeval 2016 task 11 BIBREF25.

DepTreeDepth: maximum depth of the dependency tree of the source divided by that of the target (we do not feed any syntactic information other than this ratio to the model). This parameter is designed to approximate syntactic complexity. Deeper dependency trees indicate dependencies that span longer and possibly more intricate sentences. DepTreeDepth proved better in early experiments over other candidates for measuring syntactic complexity such as the maximum length of a dependency relation, or the maximum inter-word dependency flux.

Experiments ::: Experimental Setting

We train a Transformer model BIBREF26 using the FairSeq toolkit BIBREF27. ,

Our models are trained and evaluated on the WikiLarge dataset BIBREF10 which contains 296,402/2,000/359 samples (train/validation/test). WikiLarge is a set of automatically aligned complex-simple sentence pairs from English Wikipedia (EW) and Simple English Wikipedia (SEW). It is compiled from previous extractions of EW-SEW BIBREF11, BIBREF28, BIBREF29. Its validation and test sets are taken from Turkcorpus BIBREF9, where each complex sentence has 8 human simplifications created by Amazon Mechanical Turk workers. Human annotators were instructed to only paraphrase the source sentences while keeping as much meaning as possible. Hence, no sentence splitting, minimal structural simplification and little content reduction occurs in this test set BIBREF9.

We evaluate our methods with FKGL (Flesch-Kincaid Grade Level) BIBREF30 to account for simplicity and SARI BIBREF9 as an overall score. FKGL is a commonly used metric for measuring readability however it should not be used alone for evaluating systems because it does not account for grammaticality and meaning preservation BIBREF12. It is computed as a linear combination of the number of words per simple sentence and the number of syllables per word:

On the other hand SARI compares the predicted simplification with both the source and the target references. It is an average of F1 scores for three n-gram operations: additions, keeps and deletions. For each operation, these scores are then averaged for all n-gram orders (from 1 to 4) to get the overall F1 score.

We compute FKGL and SARI using the EASSE python package for SS BIBREF31. We do not use BLEU because it is not suitable for evaluating SS systems BIBREF32, and favors models that do not modify the source sentence BIBREF9.

Experiments ::: Overall Performance

Table TABREF24 compares our best model to state-of-the-art methods:

BIBREF12

Phrase-Based MT system with candidate reranking. Dissimilar candidates are favored based on their

Levenshtein distance to the source.

BIBREF33

Deep semantics sentence representation fed to a monolingual MT system.

BIBREF9

Syntax-based MT model augmented using the PPDB paraphrase database BIBREF34 and fine-tuned towards SARI.

BIBREF10

Seq2Seq trained with reinforcement learning, combined with a lexical simplification model.

BIBREF17

Seq2Seq model based on the pointer-copy mechanism and trained via multi-task learning on the Entailment and Paraphrase Generation tasks.

BIBREF15

Standard Seq2Seq model. The second beam search hypothesis is selected during decoding; the hypothesis number is an hyper-parameter fine-tuned with SARI.

BIBREF35

Seq2Seq with a memory-augmented Neural Semantic Encoder, tuned with SARI.

BIBREF16

Seq2Seq integrating the simple PPDB simplification database BIBREF36 as a dynamic memory. The database is also used to modify the loss and re-weight word probabilities to favor simpler words.

We select the model with the best SARI on the validation set and report its scores on the test set. This model only uses three parameters out of four: NbChars$_{0.95}$, LevSim$_{0.75}$ and WordRank$_{0.75}$ (optimal target ratios are in subscript).

ACCESS scores best on SARI (41.87), a significant improvement over previous state of the art (40.45), and third to best FKGL (7.22). The second and third models in terms of SARI, DMASS+DCSS (40.45) and SBMT+PPDB+SARI (39.96), both use the external resource Simple PPDB BIBREF36 that was extracted from 1000 times more data than what we used for training. Our FKGL is also better (lower) than these methods. The Hybrid model scores best on FKGL (4.56) i.e. they generated the simplest (and shortest) sentences, but it was done at the expense of SARI (31.40).

Parametrization encourages the model to rely on explicit aspects of the simplification process, and to associate them with the parameters. The model can then be adapted more precisely to the type of simplification needed. In WikiLarge, for instance, the compression ratio distribution is different than that of human simplifications (see Figure FIGREF25). The NbChars parameter helps the model decorrelate the compression aspect from other attributes of the simplification process. This parameter is then adapted to the amount of compression required in a given evaluation dataset, such as a true, human simplified SS dataset. Our best model indeed worked best with a NbChars target ratio set to 0.95 which is the closest bucketed value to the compression ratio of human annotators on the WikiLarge validation set (0.93).

Ablation Studies

In this section we investigate the contribution of each parameter to the final SARI score of ACCESS. Table TABREF26 reports scores of models trained with different combinations of parameters on the WikiLarge validation set (2000 source sentences, with 8 human simplifications each). We combined parameters using greedy forward selection; at each step, we add the parameter leading to the best performance when combined with previously added parameters. With only one parameter, WordRank proves to be best (+2.28 SARI over models without parametrization). As the WikiLarge validation set mostly contains small paraphrases, it only seems natural that the parameter related to lexical simplification gets the largest increase in performance.

LevSim (+1.23) is the second best parameter. This confirms the intuition that hypotheses that are more dissimilar to the source are better simplifications, as claimed in BIBREF12, BIBREF15.

There is little content reduction in the WikiLarge validation set (see Figure FIGREF25), thus parameters that are closely related to sentence length will be less effective. This is the case for the NbChars and DepTreeDepth parameters (shorter sentences, will have lower tree depths): they bring more modest improvements, +0.88 and +0.66.

The performance boost is nearly additive at first when adding more parameters (WordRank+LevSim: +4.04) but saturates quickly with 3+ parameters. In fact, no combination of 3 or more parameters gets a statistically significant improvement over the WordRank+LevSim setup (p-value < 0.01 for a Student's T-test). This indicates that parameters are not all useful to improve the scores on this benchmark, and that they might be not independent from one another. The addition of the DepTreeDepth as a final parameter even decreases the SARI score slightly, most probably because the considered validation set does not include sentence splitting and structural modifications.

Analysis of each Parameter's Influence

Our goal is to give the user control over how the model will simplify sentences on four important attributes of SS: length, paraphrasing, lexical complexity and syntactic complexity. To this end, we introduced four parameters: NbChars, LevSim, WordRank and DepTreeDepth. Even though the parameters improve the performance in terms of SARI, it is not sure whether they have the desired effect on their associated attribute. In this section we investigate to what extent each parameter controls the generated simplification. We first used separate models, each trained with a single parameter to isolate their respective influence on the output simplifications. However, we witnessed that with only one parameter, the effect of LevSim, WordRank and DepTreeDepth was mainly to reduce the length of the sentence (Appendix Figure FIGREF30). Indeed, shortening the sentence will decrease the Levenshtein similarity, decrease the WordRank (when complex words are deleted) and decrease the dependency tree depth (shorter sentences have shallower dependency trees). Therefore, to clearly study the influence of those parameters, we also add the NbChars parameter during training, and set its ratio to 1.00 at inference time, as a constraint toward not modifying the length.

Figure FIGREF27 highlights the cross influence of each of the four parameters on their four associated attributes. Parameters are successively set to ratios of 0.25 (yellow), 0.50 (blue), 0.75 (violet) and 1.00 (red); the ground truth is displayed in green. Plots located on the diagonal show that most parameters have an effect their respective attributes (NbChars affects compression ratio, LevSim controls Levenshtein similarity...), although not with the same level of effectiveness.

The histogram located at (row 1, col 1) shows the effect of the NbChars parameter on the compression ratio of the predicted simplifications. The resulting distributions are centered on the 0.25, 0.5, 0.75 and 1 target ratios as expected, and with little overlap. This indicates that the lengths of predictions closely follow what is asked of the model. Table TABREF28 illustrates this with an example. The NbChars parameter affects Levenshtein similarity: reducing the length decreases the Levenshtein similarity. Finally, NbChars has a marginal impact on the WordRank ratio distribution, but clearly influences the dependency tree depth. This is natural considered that the depth of a dependency tree is very correlated with the length of the sentence.

The LevSim parameter also has a clear cut impact on the Levenshtein similarity (row 2, col 2). The example in Table TABREF28 highlights that LevSim increases the amount of paraphrasing in the simplifications. However, with an extreme target ratio of 0.25, the model outputs ungrammatical and meaningless predictions, thus demonstrating that the choice of a target ratio is important for generating proper simplifications.

WordRank and DepTreeDepth do not seem to control their respective attribute as well as NbChars and LevSim according to Figure FIGREF27. However we witness more lexical simplifications when using the WordRank ratio than with other parameters. In Table TABREF28's example, "designated as" is simplified by "called" or "known as" with the WordRank parameter. Equivalently, DepTreeDepth splits the source sentence in multiple shorter sentences in Table FIGREF30's example. More examples exhibit the same behaviour in Appendix's Table TABREF31. This demonstrates that the WordRank and DepTreeDepth parameters have the desired effect.

Conclusion

This paper showed that explicitly conditioning Seq2Seq models on parameters such as length, paraphrasing, lexical complexity or syntactic complexity increases their performance significantly for sentence simplification. We confirmed through an analysis that each parameter has the desired effect on the generated simplifications. In addition to being easy to extend to other attributes of text simplification, our method paves the way toward adapting the simplification to audiences with different needs.

Appendix

Appendix ::: Architecture details

Our architecture is the base architecture from BIBREF26. We used an embedding dimension of 512, fully connected layers of dimension 2048, 8 attention heads, 6 layers in the encoder and 6 layers in the decoder. Dropout is set to 0.2. We use the Adam optimizer BIBREF37 with $\beta _1 = 0.9$, $\beta _2 = 0.999$, $\epsilon = 10^{ -8}$ and a learning rate of $lr = 0.00011$. We add label smoothing with a uniform prior distribution of $\epsilon = 0.54$. We use early stopping when SARI does not increase for more than 5 epochs. We tokenize sentences using the NLTK NIST tokenizer and preprocess using SentencePiece BIBREF38 with 10k vocabulary size to handle rare and unknown words. For generation we use beam search with a beam size of 8.