Improving the Quality of Suggestions in a Text Simplification Tool

Jorge Aparicio, David Kauchak, Pomona College Department of Computer Science

Research Goals

This research project was divided into two distinct parts.

1. Preparing methods to gather data from the text simplification to be used in machine learning.
2. Running a short study to aid in determining the real world utility of the metric used to determine the simplification suggestions.

Both aspects of the research project were done with the purpose of making improvements on the the text simplification tool described in *A Web Based Simplification Tool* (Kauchak and Leroy). The tool aims to be broadly accessible and provide a quantifiable improvement in the understandability of texts simplified with the tool.

A major problem in the field of text simplification has always been determining whether suggestion was “good”. Earlier methods applied focused on more rudimentary factors, such as word frequency in a corpus, that were solely used to provide simplification suggestions (Shardlow).

Thus, the two sections of this project focus on determining new methods that can be used in conjunction with previous methods to provide better suggestions.

Building a Dataset for Machine Learning

Data Extraction

The first step of this process is gathering the data generated from the text simplification tool. The simplification tool used a MySQL database to store usage data from various sessions. Of the data saved by the tool, only a few different pieces of information would be useful to attain for machine learning.

The most important values determined to be used in learning were the original word, the suggested replacement, whether the suggestion was used, and the list suggestion options. Along with these data points, the session and text id were included to filter to allow for the inclusion/exclusion of certain sessions as desired.

Feature Selection

After the data is compiled, feature extraction must be performed. This is the process where the data is taken and broken down into values that can be used for learning.

- Original text features
  - Length of text
  - Number of words
- Replacement options features
  - Average length of text
  - Average number of words
- Label
  - Replacement Chosen

original_word	length	Num_word	replaced_word	r_length	r_num_words	replaced	options
poses	5	1	present	7	1	1	present,model,sit,put,set,place,
poses	5	1	model	5	1	0	present,model,sit,put,set,place,
poses	5	1	sit	3	1	0	present,model,sit,put,set,place,
poses	5	1	put	3	1	0	present,model,sit,put,set,place,
poses	5	1	set	3	1	0	present,model,sit,put,set,place,
poses	5	1	place	5	1	0	present,model,sit,put,set,place,
poses	5	1	position	8	1	0	present,model,sit,put,set,place,
poses	5	1	lay	3	1	0	present,model,sit,put,set,place,
poses	5	1	stick	5	1	0	present,model,sit,put,set,place,

Figure 1. Sample of complete data set demonstrating the features and label built with the intent of improving replacement quality.

The idea behind these features was for the classifier to determine if there is a pattern between the length and number of words in the original compared to the replacement and how likely these factors affect whether someone will accept a suggestion from the given list of suggestions.

At this stage the dataset would be complete and ready for learning, however, an important aspect of building a machine learning dataset is having enough valid data to learn from which was not yet available during the time of this project.

Determining an Effective Similarity Score

One key way the text simplification tool determines whether a suggestion will be given to the end user is on the basis of its similarity. The similarity score is calculated in a few steps. First the document as a whole is used to create vector by taking the average of all the word vectors in the document. Then the vector for the word in question is given a score by comparing it with the document vector. The same was done for each potential replacement option. Finally the score given to each option was divided by the score of the original word to give the final similarity score.

The purpose of the score is to create a system by which to rank synonyms of the original word in an attempt to exclude synonyms that do not fit the context of the passage and prioritize the presentation of the most similar synonyms.

However, it remained unclear to what degree the score was effective at filtering effective simplification suggestions from ineffective suggestions. In order to determine this cutoff we conducted a small study on Amazon Mechanical Turk, an online tool where users of the site can perform tasks for a reward.

Method

The study gave the participants a sentence containing a bolded word and an potential replacement option. The participants were given instructions to determine whether the given text could replace the bolded text in the sentence with little to no changes to the structure or meaning of the sentence.

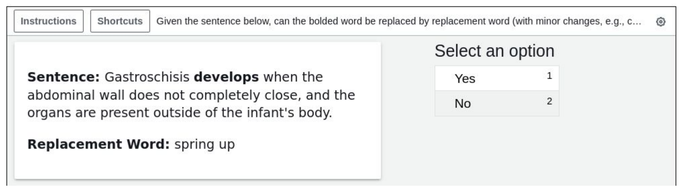

Figure 2. Sample image of a task given to a participant in the Mechanical Turk Study.

There were a total of 788 unique examples; however, each example was tested three times. Moreover, each was given three random examples to complete.

After the data from the participants were collected the answers for the duplicates of each example were tallied and the majority vote was taken to represent the example. From there the accuracy, precision, recall and F1 was calculated using different similarity scores as cutoffs.

Results & Discussion

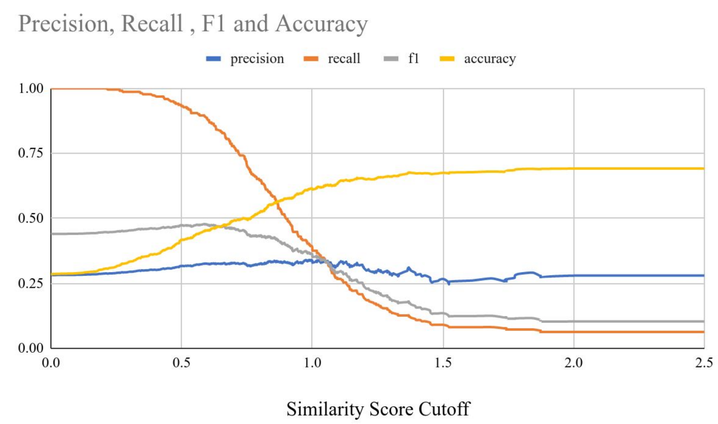

Figure 3. A graph of all the data from the Mechanical Turk study displaying how precision, recall, F1, and accuracy change with the increasing similarity score cutoff.

In order to determine what the optimal similarity score cutoff was we chose to prioritize the highest F1 score. This meant the cut off that maintained both the highest recall and precision simultaneously. Precision is the ability of our model to identify only the relevant data while recall is the ability of our model to identify all of the relevant cases within the dataset. Thus using F1 avoided using scores that maximized either precision or recall individually because both or necessary for the model to function.

The highest F1 score was 0.478 which corresponded with a similarity score cutoff of 0.587. This meant that the suggestions with a similarity score lower than about .58 tended to be of bad quality and could not substitute the original word effectively. Moreover, this also confirmed that the similarity score does have an effect in filtering out bad synonyms and prioritizing better ones.

Conclusion

In sum, at the time of the project enough data had not been collected to produce meaningful results from the machine learning classifiers. Nevertheless, as the tool is used and tested more in the future more valid data can be gathered and compiled to use to train classifiers. Once this step accomplished testing various classifiers to get a measure of performance will be possible leaving pathways towards future work on this aspect of the project.

On the other hand, the similarity score has verifiably proved that it is capable filtering out bad simplifications up to a certain extent. This proves the utility of the simplification score which can hopefully be adopted more widely as a common tool in the toolbelt for text simplification.

Citations

1. Kauchak, Leroy, “A Web Based Medical Text Simplification Tool”Shardlow,
2. Matthew. “A Survey of Automated Text Simplification.” *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 1, 2014, 10.14569/specialissue.2014.040109.