International Journal of Computational Linguistics and Applications, Vol. 6, No. 1, 2015, pp. 103–116
Received 19/02/2015, Accepted 18/05/2015, Final 14/06/2015.
ISSN 0976-0962, http://ijcla.bahripublications.com

Automated Evaluation of Short Summaries

DIEGO AGUIRRE [1]
ANTHONY MORSE [2]
OLAC FUENTES [1]

[1] *University of Texas, USA*
[2] *State University of New York, USA*

ABSTRACT

Children in elementary school are not only taught to read, but to understand what they are reading. To assess and improve their ability to understand concepts, students are often required to write short summaries of articles. Due to their nature, these documents often include misspelled words, missing punctuation, and erroneous grammatical structure. Evaluating these summaries is a laborious task that not only demands a significant amount of time from professors, but also limits the speed in which students can receive feedback. This paper presents a method for evaluating short summaries written by elementary school students. Our experiments show that incorporating semantic similarity/relatedness measures between words benefits the tasks of attribute selection and attribute weighting. We also show that preprocessing steps, such as the correction of misspelled words, are beneficial for the evaluation of short summaries. Our automatic grader has a mean absolute error of 0.98 when compared to a human grader on a 9-point grading scale. This agreement is comparable to the average agreement between two human graders.

Keywords: Natural language processing, machine learning, summary evaluation

1. INTRODUCTION

Students in elementary school are given many assignments to assess and improve their understanding of different pieces of text. In particular, young students are taught to identify the main idea of different articles along with the structure that authors use to convey ideas. For instance, students are often required to identify if the writer of an article is comparing two or more objects, or if the writer is presenting a problem and its solution. To evaluate their understanding, students are usually required to read an article on a computer and write a recall. In such recall, students are expected to state the main idea of the article along with supportive sentences that describe the structure that the author is using. Even though this process is done using a computer, professors are still required to read these summaries and grade them manually. This is a time-consuming task and makes it impossible to provide students with immediate feedback.

The automatic assessment of summaries has been studied by the text summarization community for several years. The objective is to evaluate summaries that are generated by automated tools. The methods employed usually compare fragments of the summary being evaluated against reference summaries produced by humans [4]. However, summaries written by elementary school students are different from those generated by automated tools. They are a couple of sentences long, have a significant number of misspelled words, and require a fast assessment to provide students with timely feedback. In this article, we present a method for evaluating this special type of summaries using text categorization and variable estimation techniques.

2. RELATED WORK

There has been much work in the field of automatic text summarization. A key task that researchers in this area have been studying for several years is the evaluation of automatically generated summaries. Lin and Hovy [11] addressed this problem by introducing an evaluation mechanism for this type of summaries. Their idea is based on a scoring system used for the evaluation of machine translation systems called BLEU. This scoring mechanism measures how close automatically generated translations are to translations made by humans. To do so, they use the frequencies of n-grams that are common in both machine-generated translations and reference translations. Thus, BLEU is a measure of how well an automated translation overlaps with reference translations using co-occurrences of n-grams to make the comparison. Lin and Hovy propose to use this idea to evaluate machine-generated summaries. They found that using only unigrams instead of n-grams produced better results for their task. They claim that this is caused by the fact that n-grams tend to score for grammatical structure rather than content. Their results show that using co-occurrence statistics with unigrams produces assessments that are highly correlated with human assessments [11].

A limitation with the n-gram approach is that summaries with different content can be considered equally good. This is mainly due to the fact that people express similar ideas using different words. Harnly et al. [9] propose to use what is called the automated pyramid method to address this limitation. Their method addresses some characteristics associated with abstractive summaries. These are that summaries with the same quality not only have an overlap in content, but also have a unique contribution, and that wording to express the same content can unpredictably vary. The automated pyramid method requires having multiple reference summaries available. These summaries are used to identify text fragments, called contributors, that are believed to express the same meaning. These fragments are weighted based on their frequencies in the text objects. These contributors are used to create what is called a pyramid. When an unseen summary is evaluated, its text is compared with the pyramid of contributors to see if there are any candidate contributors in the unseen summary that express the same meaning as the contributors in the pyramid. These candidate contributors are weighted using their frequencies. Finally, the score of the summary is the ratio between the sum of the weights of its candidate contributors and the sum of weights of an ideal summary. They define an ideal summary as a summary that uses the candidates from the pyramid with the highest weights and has the same size as the summary being graded. They compared their automated version of the pyramid method with its non-automated counterpart. They used Pearson and Spearman correlations as comparison metrics. Their results produced the values 0.942 and 0.943 for these correlation measures respectively [9]. This shows that the results given by this approach are very similar to the ones produced by human graders.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is another evaluation tool that uses human summaries to measure the quality of unseen summaries [10]. This mechanism is based on n-gram co-occurrence statistics and the extraction and weighting of what they call the longest common subsequence. ROUGE has also shown to provide grades that are highly correlated with human-assigned grades.

Even though the above-mentioned methods have shown to be accurate when compared to human graders, they are limited by the availability of reference summaries. Louis and Nenkova [12] propose an automated evaluation method that does not use these human models. They propose to evaluate summaries by directly measuring how close they are to the original text. Even though they introduce different mechanisms to perform this comparison, their results show that using Jensen Shannon divergence alone as a measure of similarity between the original text and the summary leads to a 0.9 correlation with human rankings for pyramid scores. This shows that automatic essay evaluation without the use of human models is at the very least promising.

3. EXPERIMENTAL DATA

A data set consisting of 7870 summaries written by elementary school students was provided by Penn State University. The article used for the summarization process had a length of 98 words. Each of the summaries was manually graded by a specialist in a 9-point scale. The distribution of grades in the data set is presented in Table 1.

4. Approach

Our approach can be seen as a three-step process. We first preprocess the text and extract features and their values from the summaries and the original article. In the second step, we use a subset of the obtained features to create three binary classifiers that learn to separate summaries at different points in the grading scale. To train each of these classifiers, we used a threshold *t* to partition the set of summaries into two: those that had a grade smaller than the threshold, and the rest. We found that the following thresholds provided the best results: 2.5, 4.5, and 7.5. In addition to the binary classifiers, we train another classifier that computes the grade of summaries given the distribution of part-of-speech tags used in the text. As a last step, we use the outputs of these classifiers along with other text-complexity/high-level features to train final classifier. Figure 1 shows the configuration of the different classifiers and the extracted features.

Table 1. *Distribution of grades*

Grade	Num.	Instances Percentage
1	190	2.41%
2	3832	48.69%
3	111	1.41%
4	388	4.93%
5	99	1.26%
6	1838	23.35%
7	412	5.24%
8	648	8.23%
9	352	4.48%

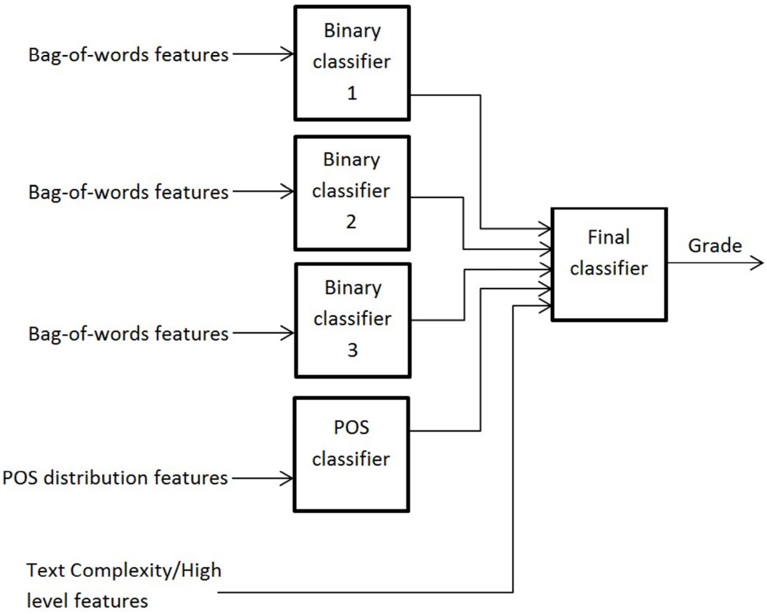

Figure 1. *Summary grading overall process*

4.1. *Data preprocessing*

Our dataset is composed of summaries that have a significant amount of misspelled words. We found that running a spell checker before extracting features from the text objects produced better results. We used the *Jazzy* library to Automated Evaluation of Short Summaries 5 programmatically replace all misspelled words in the summaries with correctly spelled ones. When a misspelled word was identified, a list of correctly spelled words was generated by the library. We replaced the misspelled word with the first suggestion, unless one of the other suggested words was a word used in the original article. In such case, we used the word found in the article to replace the misspelled word.

Additionally, we used Stanford's coreference resolution system to find all expressions that refer to the same entity in each summary and the original article. We replaced all mentions to an entity with the text that was first used when the entity was introduced in the text. For example, consider the following text extract: *John loves to go mountain biking. He enjoys being outdoors*. The resulting text after out preprocessing stage would be the following: *John loves to go mountain biking. John enjoys being outdoors*.

4.2. *Binary classifiers*

We trained three classifiers that learned to segment the dataset into two. The first classifier learned to identify the summaries that had a grade greater than or equal to 2.5 from the rest. Similarly, the other two classifiers learned to partition the dataset dataset into two with 4.5 and 7.5 as the separating grades. We tested different learning algorithms to train these binary classifiers. We found that Multinomial Naive Bayes (MNB) and Support Vector Machines (SVM) were the ones that produced the best results.

For this binary classification task, we formed a bag of words to describe the elements in the dataset. We tried two approaches for selecting the words that would form the bag. In the first approach, we used all of the non-stop words in the original article that the students summarized. Notice that the bag was not formed using the set of words that the students used in their summaries. We found that using the words from the original article to form the bag produced better results. In the second approach, we extended the number of features by also incorporating all the bigrams that could be formed using all of the words in the original article. We used the TF-IDF measure to weight the attributes in the experiments where the SVM classifier was used as the binary discriminator. For the MNB classifier, we used the frequencies of the words as weights since the algorithm is designed to work with such frequencies.

We noticed that some summaries referred to the same concepts that the original article covered. However, the words that the students used to describe these concepts were not the same as the ones used by the author of the original article. As a result, we incorporated semantic and relatedness measures to influence how the frequencies of the words in the bag for a given summary are computed. The following pseudo-code describes how the frequencies of the words in the bag are computed for a given summary *d*.

Algorithm 1: Formation of set T

```
for every word $w_o$ in the bag of words
{
   freq$_{w_o}$:= 0
   for every word $w_d$ in summary $d$
   {
      if $w_d$ = $w_o$ OR sim($w_d$, $w_o$) $>$ 0.9
      {
         freq$_{w_o}$:= freq$_{w_o}$ + 1
      }
   }
}
```

To determine the similarity between two words, we used the adapted Lesk measure found in theWordNet: Similarity library. Lesk [1] proposed that the similarity of two words is proportional to the extent of the overlaps of their dictionary definitions. Banerjee and Pedersen [7] improved on this work by incorporating WordNet as the dictionary used for the word definitions. This similarity notion was improved once more by incorporating the network of relationships between concepts in WordNet. The implementation of this adapted Lesk measure is found in the WordNet::Similarity library.

4.3. *POS classifier*

We incorporated a classifier to estimate the grade of a summary given the distribution of the part-of-speech tags that it uses. To do this, we used Stanford's NLP library to extract all part-of-speech tags from summaries. We counted the frequencies of each possible tag for each summary. We normalized the information and used the distributions as vector representations of the summaries. The classifier that gave the best results for this task was a feed-forward neural network. The number of epochs used was 4000. The learning rate was set to 0.1 and the momentum was given a value of 0.5. The number of hidden layers was 3.

4.4. *Text-complexity/high-level features*

The following features were used in combination with the binary and POS classifiers to characterize each summary in the dataset.

- Number of words in the summary
- Number of sentences
- Number of misspelled words
- Average word length
- Euclidean distance between the original article and the summary using the bag of words weights as descriptors
- Percentage of words in the original article that appear in the summary
- Number of words longer than 5 characters
- Number of words longer than 6 characters
- Number of words longer than 7 characters

The output of the binary and POS classifiers along with the above-mentioned features were given as input to a final classifier. We tried two types of classifiers for this last step: A feedforward neural network and k-nearest neighbors. The number epochs used for the neural network was 4000. The learning rate was set 0.1 and the momentum was given a value of 0.5. The number of hidden layers was 3. For k-nearest neighbors, we found that k=3 produced the best results. The output of this final classifier was rounded since the grades assigned to the summaries are discrete.

5. EVALUATION

We are interested in comparing the grades given by our system to the ones assigned by the human grader. The following three metrics allow us to analyze this from different perspectives.

- Mean absolute error (MAE)
- Exact (E): number of summaries that were given the same grade as the human grader over the total number of summaries

- Adjacent (A): number of summaries that were given a grade that differed from the human grade by 1 point over the total number of summaries

To assess our approach, the data set was randomly split into two equal-sized subsets, preserving the distribution of the original grades. One of these two sets was used for training and the other for testing. Since our approach cannot not be directly compared to other approaches due to the uniqueness of the dataset, we developed a baseline approach where all extracted features were used for training. That is, a single classifier was trained using all features extracted from the training set and evaluated using the testing set. Comparing our approach to this simple baseline model allows us to recognize and approciate the utility of our summary evaluation system.

6. RESULTS

Table 2 shows the results obtained by our baseline. For each experiment, all attributes were used for training (BOW, POS, and text-complexity/high-level features). The column *Classifier* indicates the type of classifier used as the baseline. *BOW features* indicates how the bag-of-words was constructed. MAE indicates the mean absolute error of the experiment. *Exact* indicates the percentage of summaries that were given the same grade as the grade assigned by the human grader. *Adjacent* indicates the percentage of summaries that were given a grade that differed from the human grade by only 1 point.

Table 2. *Baseline results*

Classifier	**BOW features**	**MAE**	**Exact**	**Adjacent**
FFNN	Non-stop words	1.26	.44	.64
FFNN	Non-stop words + bigrams	1.28	.43	.66
SVM Regression	Non-stop words	1.17	.31	.69
SVM Regression	Non-stop words + bigrams	1.20	.30	.68
KN	Non-stop words	1.27	.38	.65
KNN	Non-stop words + bigrams	1.29	.35	.63

Table 3. *Proposed approach results*

Binary Classifier	BOW features	Final Classifier	MAE	Exact	Adjacent
MNB	Non-stop words	FFNN	0.98	.43	.77
MNB	Non-stop words + bigrams	FFNN	0.99	.42	.75
SVM	Non-stop words	FFNN	1.14	.35	.69
SVM	Non-stop words + bigrams	FFNN	1.16	.34	.68
MNB	Non-stop words	KNN	1.3	.35	.63
MNB	Non-stop words + bigrams	KNN	1.33	.35	.64
SVM	Non-stop words	KNN	1.4	.32	.65
SVM	Non-stop words + bigrams	KNN	1.41	.33	.66

Table 3 shows the results obtained when using binary and POS classifiers in combination with text-complexity/high-level features. Each row in the table represents an experiment. The column labeled *Binary Classifier* indicates the type of classifier that was used to partition the data set at the three different points in the grading scale. *BOW features* indicates what features were used to train the binary classifiers. *Final Classifier* indicates the type of classifier that was to ultimately estimate the grade of a summary. The columns *MAE*, *Exact*, and *Adjacent* have the same meaning as for Table 2.

7. Discussion

Different conclusions can be drawn from the results. We observe that the feed forward neural network outperforms k-nearest neighbors in all instances. This can be easily attributed to the fact that neural networks, although not always, tend to outperform algorithms such as k-nearest neighbors in many estimation problems. It is also interesting to notice that the incorporation of bigrams did not have a significant effect in the obtained results. We expected the incorporation of bigrams to have a positive effect on all of the described metrics since other natural language tasks have been benefited from such process. We attribute this to the nature of the problem we are solving. In other tasks, such as sentiment analysis, words such as “not” and “no” play a very important role. Thus, bigrams where these words appear tend to be appropriate attributes for the text objects. In our problem, students are meant to identify and write the main concepts that an article presents. Thus, using single words to form the bag suffices for the problem at hand.

We also observe that MNB is a suitable classifier for partitioning the dataset into 2 at different points in the grading scale. Although SVMs are suitable for binary classification problems, MNB showed to be a better candidate for this task. This same result has been observed in other problems where the classification problem involves text objects.

The results also show that our approach outperforms the baseline in terms of MAE using the best configuration that we found. More concretely, our best result was 0.98 in contrast to 1.17 from the baseline (a significant difference of almost 0.2). For the Adjacent metric, we also observe a remarkable difference between the two approaches, showing the robustness of the proposed approach. However, the baseline appeared to perform just as well as our approach for the Exact metric. In conclusion, we see that the combination of binary classifiers, a POS classifier, and other text-complexity/high-level features to train a final classifier outperforms the traditional approach of using all features to train a single classfier.

Finally, the results show that grading short summaries is a task that can be performed by a computer system reliably. In our best result, we see that our solution gave a grade that differed by at most one point from the actual grade for 77% of the cases. A one-point difference is something that is expected even when comparing two human graders. Although the grading of this type of summaries might still require human intervention to provide students with more concrete and detailed feedback, our solution can be of great use to quickly assess the quality of summaries written by young students. This can aid students when typing their summaries on a computer. Our solution can quickly analyze the text and provide students with early feedback that they can use to improve their summaries before submitting them to the professor.

REFERENCES

1. Banerjee, S. & Pedersen, T. 2002. An adapted lesk algorithm for word sense disambiguation using wordnet. In *Computational Linguistics and Intelligent Text Processing* (pp. 136-145). Springer.
2. Cavnar, W. B. & Trenkle, John M. et al. 1994. N-gram-based text categorization. *Ann Arbor MI*, 48113(2), 161-175.
3. Feldman, R. & Sanger, J. 2007. The text mining handbook: advanced approaches in analyzing unstructured data. Cambridge University Press.
4. Hovy, E., Lin, C-Y., Zhou, L. & Fukumoto, J. 2006. Automated summarization evaluation with basic elements. In proceedings of *the Fifth Conference on Language Resources and Evaluation* (LREC 2006), (pp. 604-611). Citeseer.
5. Lewis, D. D. 1992. Feature selection and feature extraction for text categorization. In proceedings of the workshop on Speech and Natural Language (pp. 212-217). Association for Computational Linguistics.
6. Manning, C. D. 2011. Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In *Computational Linguistics and Intelligent Text Processing* (pp. 171-189). Springer.
7. Pedersen, T., Patwardhan, S. & Michelizzi, J. 2004. Wordnet: Similarity: Measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004* (pp. 38-41). Association for Computational Linguistics.
8. Yang, Y. & Pedersen, J. O. 1997. A comparative study on feature selection in text categorization. In ICML, 97, 412-420.
9. Harnly, A., Nenkova, A., Passonneau, R. & Rambow, O. 2005. Automation of summary evaluation by the pyramid method. In *Recent Advances in Natural Language Processing (RANLP)*, (pp. 226-232).
10. Lin, C-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop* (pp. 74-81).
11. Lin, C-Y. & Hovy, E. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In proceedings of the *2003 Conference of the North American Chapter of the Association for*

Computational Linguistics on Human Language Technology (pp. 71-78), Vol. 1, Association for Computational Linguistics.

12. Louis, A. & Nenkova, A. 2008. Automatic summary evaluation without human models. In *Notebook Papers and Results, Text Analysis Conference (TAC-2008),* Gaithersburg, Maryland (USA).

DIEGO AGUIRRE
UNIVERSITY OF TEXAS AT EL PASO,
EL PASO, TX 79902, USA.
E-MAIL: <DAGUIRRE6@MINERS.UTEP.EDU>

ANTHONY MORSE
STATE UNIVERSITY OF NEW YORK AT BROCKPORT,
BROCKPORT, NY 14420, USA.
E-MAIL: <ANTHONYMORSE92@GMAIL.COM>

OLAC FUENTES
UNIVERSITY OF TEXAS AT EL PASO,
EL PASO, TX 79902, USA.
E-MAIL: <OFUENTES@UTEP.EDU>