Automatic learner summary assessment for reading comprehension

Menglin Xia
Amazon Research Cambridge*
ximengli@amazon.co.uk

Ekaterina Kochmar
ALTA Institute
Computer Laboratory
University of Cambridge
ek358@cl.cam.ac.uk

Ted Briscoe
ALTA Institute
Computer Laboratory
University of Cambridge
ejb@cl.cam.ac.uk

Abstract

Automating the assessment of learner summaries provides a useful tool for assessing learner reading comprehension. We present a summarization task for evaluating non-native reading comprehension and propose three novel approaches to automatically assess the learner summaries. We evaluate our models on two datasets we created and show that our models outperform traditional approaches that rely on exact word match on this task. Our best model produces quality assessments close to professional examiners.

1 Introduction

Summarization is a well-established method of measuring reading proficiency in traditional English as a second or other language (ESOL) assessment. It is considered an effective approach to test both cognitive and contextual dimensions of reading (Weir et al., 2013). However, due to the high time and cost demands of manual summary assessment, modern English exams usually replace the summarization task with multiple choice or short answer questions that are easier to score (Alderson, 2005). Automating the assessment of learner summarization skills provides an efficient evaluation method for the quality of the learner summary and can lead to effective educational applications to enhance reading comprehension tasks.

In this paper, we present a summarization task for evaluating non-native reading comprehension and propose three novel machine learning approaches to assessing learner summaries. First, we extract features to measure the content similarity between the reading passage and the summary. Secondly, we calculate a similarity matrix based on sentence-to-sentence similarity between the reading passage and the summary, and apply a Convolutional Neural Network (CNN) model to assess the summary quality using the similarity matrix. Thirdly, we build an end-to-end summarization assessment model using the Long Short Term Memory (LSTM) model. Finally, we combine the three approaches in a single system using a simple parallel ensemble modeling technique. We compiled two datasets to evaluate our models, and we release this data with the paper. We show that our models outperform traditional approaches that rely on exact word match on the task and that our best model produces quality assessments close to professional examiners.

2 Related Work

Most of the previous studies on summary assessment are aimed at evaluating automated summarization systems (Lin, 2004; Lin and Hovy, 2003; Nenkova et al., 2007). In contrast to this line of work, our goal is to assess human-written summaries rather than machine-generated ones.

Within the educational domain, several applications have been developed to help students with their writing summarization skills. Summary Street (Wade-Stein and Kintsch, 2004) is an educational software designed for children to develop summarization skills. It asks students to write a summary to a reading passage, and scores the summary by using Latent Semantic Analysis (LSA) to construct semantic representations of the text. This system uses the cosine similarity score based on LSA as the sole indicator of content similarity. OpenEssayist (Whitelock et al., 2013) is an interactive system that provides students with the automated feedback about their summative essays. The system extracts the key words, phrases and sentences from the essays and helps the students to investigate their distribution in text and the po-

*The work by the first author was done at the University of Cambridge prior to joining Amazon Research.

Proceedings of NAACL-HLT 2019, pages 2532–2542
Minneapolis, Minnesota, June 2 - June 7, 2019. ©2019 Association for Computational Linguistics

tential implications for the clarity of the narrative.

The work that is most similar to ours is the automatic scoring of a summarization task by Madnani et al. (2013), who designed a task to measure the reading comprehension skills of American students. In their experiments, students were asked to write a four-sentence summary for each of the two three-paragraph reading passages, with the first sentence summarizing the whole passage and the following three sentences summarizing each paragraph. To build an automated system to score the summaries, they randomly select a student summary with the highest score as the reference, and use 8 feature types to train a logistic regression classifier to predict the summary score. They train a separate classifier for each of the two passages, and obtain accuracy scores of 65% and 52% respectively, outperforming the most-frequent-score baselines of 51% and 32%. Most of the features used in Madnani et al. (2013) are based on verbatim overlap. Although such metrics prove to be effective in various tasks, they cannot capture the content similarity when paraphrasing or a higher level of abstraction are used in the summary.

Few studies have addressed summarization assessment at a higher level. More recently, Ruseti et al. (2018) have used Recurrent Neural Networks (RNNs) to automatically score summaries. In their model, a concatenated representation of the summary and the text built from two separate RNNs as well as a complexity score of the text are fed to a fully connected layer to predict a real number between $[0, 1]$. This number is then mapped to 3 discrete classes representing the quality of the summary using linear regression. Their best model achieves 55.24% in accuracy on a dataset of 636 summaries collected using Mechanical Turk.

In this paper, we address several limitations of previous work. We build a system that uses verbatim features as well as features capturing higher level of abstraction. First, we aim to build a generic system that can evaluate the quality of a summary without having to train a separate model for each text. Second, whereas Madnani et al. (2013) use a student summary with the highest score as the reference to evaluate the candidate summary, our goal is to build a fully-automated system that does not require selecting a pre-defined reference. Third, we aim to explore features and structures capable of better modeling semantic similarity beyond verbatim overlap.

3 Data

This section outlines the summarization task used in our experiments. First, learners, regardless of their proficiency level, were asked to read three reading passages extracted from the Cambridge English Exams dataset of Xia et al. (2016) at the lower (B1), upper intermediate (B2), and advanced (C1) levels of the Common European Framework of Reference for Languages (CEFR). Then they were asked to write a summary of 50, 100, and 120 words for each of the three tasks.[1]

3.1 Pilot study with simulated learner data

Before launching the experiments with the actual language learners and in order to develop the automated summary evaluation system, we first ran a pilot study and collected "simulated learner" summaries from 50 members of our university. Since most participants of this study would speak English at an advanced (C1-C2) level, we asked them to write a "good summary" and a "bad summary" for each reading passage, mimicking a learner. The participants were asked to produce grammatically correct sentences and to write a "bad" summary in a way that a learner who does not fully understand the original passage might produce. In total, 300 summaries were collected (with 150 good summaries and 150 bad ones). The simulated learner data was then used to train binary classification systems to assess whether a summary captures the passage content properly or not.

3.2 Real learner data

Learner levels	Count
B1	40
B2	40
C1-C2	57
Total	137

Table 1: The distribution of the learner proficiency levels in the real learner data

Next, we collected summaries from second language learners at B1, B2 and C1-C2 levels of proficiency.[2] In total, 411 summaries from 137 learners were collected. The distribution of the learner proficiency levels is shown in Table 1.

[1] The word limits on the summarization tasks are set to keep a relatively constant compression ratio between the summary and the length of the original passage.

[2] The proficiency levels of learners were self-identified.

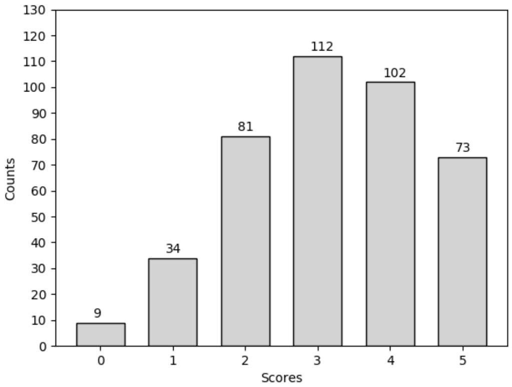


Figure 1: A histogram illustrating the score distribution in the real learner data

The summaries were then scored by three professional ESOL annotators using a 6-point scale defined as follows:

Band 5: *The summary demonstrates excellent understanding of the original passage: Content covers all the main points of the passage. All content included is accurate, with no irrelevant details or repetitions. Target reader is fully informed.*

Band 4: *Performance shares features of Bands 3 and 5.*

Band 3: *The summary demonstrates acceptable understanding of the passage: Most of the main points are included. Most of the content is relevant and paraphrased, with some irrelevant details, repetitions or inaccuracy of content. Target reader is on the whole informed.*

Band 2: *Performance shares features of Bands 1 and 3.*

Band 1: *The summary demonstrates very little understanding of the passage: Most of the content is of limited relevance, with repetitions or verbatim borrowing from the original text. In some paraphrased parts of the text, inaccuracy of content or omissions of main points are evident. Target reader is minimally informed.*

Band 0: *No understanding of the passage is demonstrated. The content is totally irrelevant to the original passage. Target reader is not informed.*

Figure 1 shows the distribution of the scores for the learner summaries. The pairwise correlation between annotators ranges between 0.690 and 0.794. To derive the final score for each summary, we take the average of the scores by the three annotators. This results in a set of real-valued average scores on the scale of $[0, 5]$ and allows us to treat this task as a regression problem and make use of the continuity of the assessment scale. The goal of the experiments on this data is then to train a regression model to predict a score that correlates well with the annotators' judgments.

4 Methods

In this section, we introduce three different approaches to the automated evaluation of the learner summaries.

4.1 Measures for summary assessment

First of all, we extract a number of features to describe the similarity of the summary and the reading text and apply a machine learning model to predict the summary quality.

The summarization task for reading comprehension examines the content relevance and the ability to convey the main ideas of the text in the summary. To automatically assess the learner summary, we compare the candidate summary against a reference to assess the quality of its content.

We experiment with two types of references to evaluate the candidate summary: firstly, we compare the candidate summary against the *original passage* directly, and secondly, we extract key sentences from the original text with an automated extractive summarizer and compare the candidate summary to the set of key sentences. Ideally, an extractive summarizer extracts a subset of sentences from the passage that are highly representative of the original text. Although the extracted key sentences are not necessarily coherent among themselves, they provide a representation of the main ideas of the text. Comparing the candidate summary against the key sentences allows us to examine the content relevance and the coverage of the main ideas in the candidate summary. We compare two popular summarizers in selecting the key sentences for reference: *TextRank* (Mihalcea and Tarau, 2004) and *MEAD* (Radev et al., 2004). We also compare the extractive summarizers against the baseline of using a *random selection* of sentences as the reference.

After obtaining the reference, we derive four types of linguistic features to evaluate the quality of the learner summary: (1) verbatim features, (2) semantic similarity features, (3) features based on distributed vector representations of the summary, and (4) features that describe discourse and other textual characteristics of the summary.

4.1.1 Verbatim features

Verbatim similarity is the most straightforward measure that indicates content similarity. Verbatim features measure the lexical overlap of the text units between the candidate summary and the reference. We use the following metrics to measure verbatim similarity: **ROUGE** (Lin, 2004), **BLEU** (Papineni et al., 2002), and **METEOR** (Denkowski and Lavie, 2011). The three metrics are commonly used to assess automated summarization systems. ROUGE and BLEU are based on exact word match of N-grams, and METEOR extends the exact word match with stem, synonym, and paraphrase matches extracted from the WordNet (Miller, 1995) and a background dictionary, which allows for more flexible expressions.

4.1.2 Semantic similarity features

Although verbatim overlap metrics prove to be effective in various tasks, they fail to capture the content similarity when paraphrasing and higher levels of abstraction are used in the summary. To compensate for this, word embeddings and sentence embeddings are used to model semantic similarity at the word and the sentence level. We measure the semantic similarity between words and sentences in the texts and combine the scores into a measure of document-level semantic similarity.

1. **Word similarity**: **Word2vec** (Mikolov et al., 2013) is a model for learning distributed vector representations of words from a large corpus of text. We use embeddings pre-trained on Wikipedia to compute word-to-word cosine similarity between the candidate summary and the reference. We experiment with three scoring functions to construct the text-level semantic similarity measures from the word-to-word scores:

   (1) *average word similarity* on every word pair in the candidate summary and the reference;

   (2) *a greedy method* (Mihalcea et al., 2006) that finds the best-matching word with maximum similarity scores and computes the average over the greedily selected pairs;

   (3) *optimal matching* (Rus and Lintean, 2012) that finds the optimal alignment of word pairs and then takes the average over the alignment.

2. **Sentence similarity**: **Skip-thought** (Kiros et al., 2015) is a model for learning distributed representations of sentences. It uses an RNN-encoder to compose the sentence vector, and a decoder conditioned on the resulting vector that tries to predict the previous and the next sentences in the context. We use the model pre-trained on the BookCorpus (Zhu et al., 2015) to generate our sentence vectors. Additionally, we experiment with composing the sentence vectors using word embedding summation and taking the average (**average word embeddings**). We use the same functions for word-level similarity to compute the text semantic similarity from the sentence vectors.

4.1.3 Distributed vector representations of the summary

In addition to the word and sentence similarities, we investigate methods to model the content similarity between the candidate summary and the reference directly at the document level.

Specifically, we use the following five approaches to construct vector representations of learner summaries:

TF-IDF is a common method to construct document representations in information retrieval. TF-IDF weighted document vectors are frequently used for measuring query-document similarity.

Doc2Vec (Le and Mikolov, 2014) is a neural network model for learning distributed representation of documents. We use the "distributed memory of paragraph vectors (PV-DM)" variant of the model to construct our vector representation of the summary. The PV-DM model maps the document to a vector space and uses a combination of the document vector and the vectors of surrounding words to predict a target word.

Latent Semantic Analysis (LSA) (Landauer, 2006) applies singular value decomposition (SVD) on the term-document matrix to obtain vector space representation of documents.

Latent Dirichelet Allocation (LDA) (Blei et al., 2003) represents the documents as mixtures of topics. It can be used to measure the content similarity and topical relevance of documents.

We also make use of the **average word embeddings** to encode the summary.

We use the Simple English Wikipedia corpus[3] as our background resource to learn the document representations. The Simple English Wikipedia data is used to train the models because its documents are rendered simple for English learners. Therefore, the lexical usage and syntactic structure in Simple English Wikipedia are more similar to the summaries written by learners. We take

[3] https://simple.wikipedia.org

the cosine similarity between the candidate and the reference vectors to evaluate their similarity.

4.1.4 Discourse and other textual features

Apart from the content-based measures of the summary, the textual quality of the summary is also important for its overall quality estimation. For instance, good summaries tend to be more coherent and logically consistent. We extract **lexical chain**-based discourse measures to assess the coherence of the text. Lexical chains model the semantic relations among entities throughout the text. We implement the lexical chaining algorithm developed by Galley and McKeown (2003) and extract 7 lexical chain-based features.[4]

We also measure the following superficial textual features:

Length: Number of words in the summary.

Compression ratio: The ratio of the number of words in the summary to the number of words in the reading passage.

Type-token ratio: The ratio of the number of unique words to the total number of words in the summary.

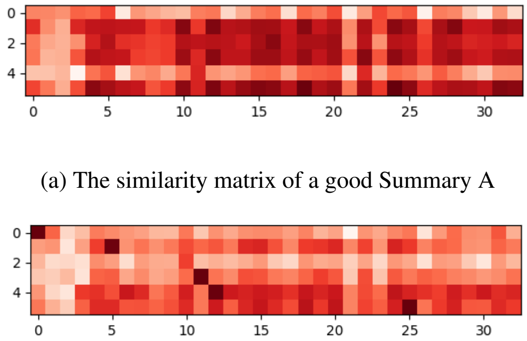
Text readability: The reading difficulty (the CEFR level) of the passage to be summarized.

After the features are extracted, we train a Support Vector Machine (SVM) (Cortes and Vapnik, 1995) model for the classification task (Section 3.1) and a Kernel Ridge Regression (KRR) (Saunders et al., 1998) model for the regression task (Section 3.2).

4.2 Assessing summary using similarity matrix

Secondly, we construct a sentence similarity matrix between the candidate summary and the original reading passage and apply a Convolutional Neural Network (CNN) model on the similarity matrix to predict the quality of the summary.

Lemaire et al. (2005) proposed a computational cognitive model for assessing extractive summarization. In their experiments, they presented 278 American school students with two reading passages and asked them to underline three to five sentences that they considered the most important in the texts. The underlined sentences were compared against the set of all the sentences from the original passage. They observed that the important sentences selected by the students are highly connected to the rest of the sentences in the text, where the connection is defined by the semantic similarity of the sentences.



(a) The similarity matrix of a good Summary A

(b) The similarity matrix of a bad Summary B

Figure 2: Similarity matrices of two summaries for the same reading passage from the simulated learner data. Summary A is a good summary and Summary B is a bad summary. The rows of the matrix represent sentences in the summary and the columns of the matrix represent sentences in the reading passage.

Based on their observations, we hypothesize that sentences in a good summary should have a well-distributed connection with as many sentences as possible in the original text, because a good summary is supposed to cover all the important information in the text. In contrast, sentences in a bad summary may fail to form a well-distributed connection with sentences in the original text. For example, if a bad summary only captures a few of the main points in the original text, then the sentences in such a summary would be connected only to the sentences where these points are mentioned in the original text, lacking the connections to the rest of the text. If a bad summary is generally irrelevant to the original text, sentences in such a summary would be minimally connected to most of the sentences in the original text. Beside these extreme cases on summary quality scale, summaries of intermediate quality may display patterns of connection to the original passage that share the characteristics of the good summary and the bad summary to various degrees.

Following this idea, we construct a sentence similarity matrix between the candidate summary and the original text. Each element of the matrix is a cosine similarity score between the vector representations of a sentence from the summary and a

[4] Features include: number of lexical chains per document, number of lexical chains normalized by text length, average/maximum lexical chain length, average/maximum lexical chain span, and the number of long chains.

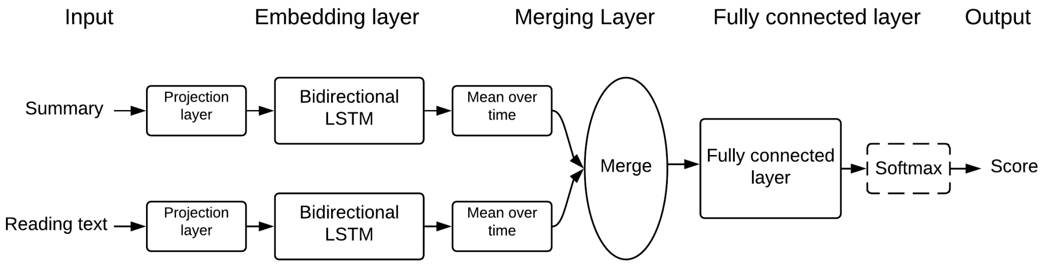


Figure 3: The merged LSTM model

sentence from the original text. We use the two sentence similarity models described in Section 4.1.2, skip-thought and average word embeddings, to build the sentence vectors.

According to our hypothesis, the quality of the summary corresponds to different patterns in the similarity matrix. The similarity matrix can be viewed as a heat map "image" from which we can learn patterns to detect the quality of the summary. Figure 2 demonstrates the similarity matrices of two summaries for the same reading passage from the simulated learner data. The shade of the coloured map indicates the degree of similarity between two sentences: the darker the shade is, the more similar the sentences are. In this example, Summary A is an example of a good summary, and Summary B is an example of a bad summary. We can see that sentences in Summary A are similar to a number of sentences in the original text, resulting in a well-distributed heat map. By contrast, sentences in Summary B are similar to five particular sentences in the text and are less similar to other sentences, which is reflected by the isolated dark patches in the image. On the whole, Summary A has higher similarity scores than Summary B, which makes its heat map darker. These two examples illustrate how different patterns may be observed in the heat map of the summaries of different quality.

To learn these patterns automatically, we apply a CNN model on the similarity matrix to predict the quality of the summary. However, it should be noted that CNNs usually work best when a large amount of training data is available, whereas the summary data we have collected represents a relatively small dataset. We compare the results of the CNN model against the feature extraction approach to investigate how well the model can learn from the limited amount of data.

4.3 Assessing summary using LSTM-based models

Thirdly, we experiment with several LSTM-based neural network models for assessing the summary quality. The LSTM-based models are used to learn representations of the summary and estimate its quality automatically, without having to manually extract features from it.

Recurrent neural networks with LSTM units (Hochreiter and Schmidhuber, 1997) have shown impressive results on various NLP tasks (Wang and Jiang, 2016; Rocktäschel et al., 2015). In essence, they are capable of embedding long text sequences into a vector representation which can later be decoded for use in various applications.

4.3.1 Merged LSTM model

Inspired by the recent advances with LSTMs in NLP tasks, we propose a merged LSTM model (see Figure 3) for assessing learner summaries. The merged LSTM model encodes the summary and the reading text separately with two bidirectional LSTMs, and merges the embedded summary and embedded reading text representations into a joint representation to predict the summary score. We explore four functions to merge the encoded vectors, including *concatenation*, *addition*, *dot product* and *linear combination*.

4.3.2 Attention-based LSTM model

As the merged LSTM model encodes the summary and reading text separately, it needs to propagate dependencies over long sequences to compare the summary and the text. The joint representation obtained in the merged LSTM model cannot fully capture the connection between the summary and the text. In this section, we propose an attention-based LSTM model which makes use of an attention mechanism to better model the relation between the summary and the reading text.

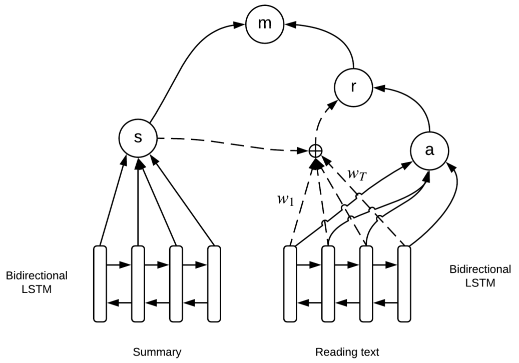


Figure 4: Attention mechanism architecture in the attention-based LSTM model for summary assessment

The attention mechanism was first introduced by Bahdanau et al. (2014) for machine translation. In general, the attention model learns a soft alignment between the input and the output in the encoder-decoder framework. The attention mechanism allows the model to learn what to attend to in the input states and mitigates the long-dependency bottleneck of the LSTM.

In the attention-based model for summary assessment, the original text and the summary are still encoded using two separate LSTMs. However, the text representation is formed by a weighted sum of the hidden states of the text encoder, where the weights can be interpreted as the degree to which the summary attends to a particular token in the text. The summary representation and the text representation are combined with a nonlinear function into a joint representation and then fed into the fully connected layer to predict the summary quality. Figure 4 is an illustration of the attention mechanism between the embedded summary and the embedded input text.

Mathematically, suppose s is the encoded summary vector and $a(t)$ is the hidden state of the LSTM for the text at each token t. Then the final representation r of the encoded text is a weighted sum of $a(t)$:

$$r = a \cdot w = \sum_{t=1}^{T} a(t)w(t)$$

The weight for each token $w(t)$ is computed by:

$$w(t) = \frac{exp(\alpha(t))}{\sum_{t=1}^{T} exp(\alpha(t))}$$

where

$$\alpha(t) = W_{a\alpha} \cdot a(t) + W_{s\alpha} \cdot s$$

is an alignment model.

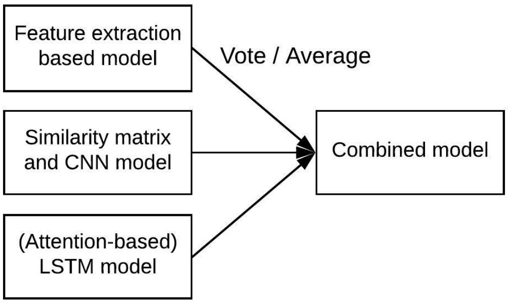

Feature extraction based model
Similarity matrix and CNN model
(Attention-based) LSTM model
Vote / Average
Combined model

Figure 5: Combining three approaches using ensemble modelling

The joint representation m of the summary and the text is a combination of the summary vector s and the weighted input text vector r.

$$m = tanh(W_{sm} * s + W_{rm} * r + b)$$

where W_{sm}, W_{rm} and b are the parameters of a linear combination function.

4.4 Ensemble modelling

Ensemble modelling combines several machine learning techniques into one model in order to improve the stability and accuracy of the prediction. We explore combining the three different models (see Figure 5) into a single model by taking the majority vote from the binary classification models and taking the average value of the predicted scores from the regression models. We compare the performance of the combined models against the individual models to investigate if and to what extent ensemble modelling is useful for assessing the summaries.

5 Experiments and Results

5.1 Experimental set-up

We evaluate our models on the real learner data and on the simulated learner data, for consistency, using 5-fold cross validation. In each fold, 60% of the data is used as the training set, 20% as the development set, and 20% as the test set.[5]

We compare our models against five baselines: *most frequent baseline*, *random baseline*, *ROUGE*

[5] We choose the best model based on the development set, retrain the selected model on the combination of the training and the development data, and evaluate the model on the test set.

Models	Variants			Accuracy
Baseline	Baseline type		most-frequent	50.0%
			random	50.0%
			ROUGE	59.3%
			BLEU	51.7%
			ROUGE + BLEU	59.6%
SVM	reference type		random	58.8%
			TextRank	63.8%
			MEAD	62.9%
			original text	**65.6%**
CNN	similarity matrix type		avg word embeddings	**65.8%**
			skip-thought vectors	63.4%
LSTM	Merged LSTM	merging function	concatenation	68.0%
			addition	68.1%
			multiplication	69.1%
			linear combination	70.4%
	Attention LSTM			**71.1%**
Combined model	SVM+CNN+LSTM			**75.3%***

Table 2: Model performance on the simulated learner data. We use the bold font to indicate the best model for each method. The asterisk sign indicates the best performance across all models.

baseline,[6] *BLEU baseline*, and *ROUGE + BLEU baseline*.

We use accuracy to evaluate the models on the simulated learner data, and on the real learner data, we report scores of two evaluation metrics: Pearson correlation coefficient (PCC) and Root Mean Square Error (RMSE), which are commonly used for evaluating regression models.

5.2 Results

Table 2 shows the results of the baseline and the four types of models on the simulated learner data, and Table 3 reports the results of the models on the real learner data.

On the simulated learner data, the best variants from all three methods outperform the baselines. The improvement is statistically significant ($p<0.05$) using t-test for all three methods. We combine the best variants from the three approaches into a single system by taking the majority vote from the models. The resulting system achieves the best accuracy of 75.3% in predicting the binary type of the summary on the simulated learner data.

On the real learner data, we found that the feature extraction-based model outperforms the CNN model and LSTM-based models, which also significantly outperform the baselines. The results suggest that the neural network-based models are not as effective as the traditional feature extraction-based method for the regression task, at least when the training data is limited in size.

However, although the CNN and LSTM models are not the best-performing models individually, a combination of the three methods (KRR, CNN and LSTM) still improves the performance. We believe that this is because the three independent models capture different aspects of the summary quality that are complementary to each other. In addition, the combined model is more robust to outliers. For example, when two models agree on an instance while the third model does not, the combined model will select the majority vote or the average score of the model estimations, hence achieving a better performance in estimating the summary quality. Overall, the best model performance is close to human performance.

We also observe that when assessing the summaries with extracted features, using the original document as the reference works better than using other types of reference. This might be because the extractive summarizers only select sentences that are highly related to others, where the relation

[6] A baseline trained on ROUGE features only.

Models	Variants			PCC	RMSE
Baseline	Baseline type		most-frequent	-	1.30
			random	0.011	1.79
			ROUGE	0.499	1.12
			BLEU	0.208	2.88
			ROUGE + BLEU	**0.499**	**1.11**
KRR	reference type		random	0.517	1.11
			TextRank	0.576	1.06
			MEAD	0.557	1.08
			original text	**0.636**	**0.99**
CNN	similarity matrix type		avg word embeddings	**0.504**	**1.12**
			skip-thought vectors	0.458	1.14
LSTM	Merged LSTM	merging function	concatenation	0.487	1.13
			addition	0.466	1.13
			multiplication	0.490	1.12
			linear combination	0.484	1.13
	Attention LSTM			**0.494**	**1.12**
Combined model	KRR+CNN+LSTM			**0.665***	**0.97***

Table 3: Results of the regression model performance on the learner data. We use the bold font to indicate the best model for each method. The asterisk sign indicates the best performance across all models.

is typically judged by the word overlap, therefore missing the bits of text where topical words occur less often.

6 Conclusion

In this paper, we introduce a summarization task for testing reading comprehension of learners and present several automated systems to assess the quality of the learner summary. We collected summaries from members of our university and from the real learners to evaluate our systems. We propose and compare three approaches to assess the summaries, including the feature extraction-based model, the CNN-based model using similarity matrix, and the LSTM-based model. The best system, built using a combination of three models, yields an accuracy of 75.3% on the simulated learner data, and $PCC = 0.665$, $RMSE = 0.97$ on the real learner data. Although not directly comparable to other studies, we note that these results are higher than those reported in previous work.

Our systems are generalizable and address the limitations of the previous research in this area as: (1) they are capable of evaluating the quality of a summary without the need of being trained on each input text separately, (2) they do not require a pre-defined reference, and (3) they are capable of capturing content similarity beyond verbatim overlap, taking into account paraphrasing and higher levels of abstraction.

We believe that although the application presented in this paper focuses on assessing learner summaries, these techniques may also be useful for benchmarking automated summarization systems. Evaluation of these techniques for benchmarking automated summarization systems is one direction for our future research.

We make the summary data available at https://www.cl.cam.ac.uk/~ek358/learner-summaries.html.

Acknowledgments

This paper reports on research supported by Cambridge Assessment, University of Cambridge. We also thank Cambridge Assessment for their assistance in the collection of the real learner data. We are grateful to the anonymous reviewers for their valuable feedback.

References

J. Charles Alderson. 2005. *Assessing reading*. Cambridge University Press.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 85–91. Association for Computational Linguistics.

Michel Galley and Kathleen McKeown. 2003. Improving word sense disambiguation in lexical chaining. In *International Joint Conference on Artificial Intelligence*, volume 3, pages 1486–1488.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ryan Kiros, Yukun Zhu, Ruslan R. Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, pages 3294–3302.

Thomas K. Landauer. 2006. *Latent semantic analysis*. Wiley Online Library.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, volume 14, pages 1188–1196.

Benoit Lemaire, Sonia Mandin, Philippe Dessus, and Guy Denhiere. 2005. Computational cognitive models of summarization assessment skills. In *Proceedings of the 27th Annual Meeting of the Cognitive Science Society*, pages 1266–1271.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out*, volume 8, pages 74–81. Association for Computational Linguistics.

Chin-Yew Lin and Eduard Hovy. 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: Human Language Technologies*, volume 1, pages 71–78. Association for Computational Linguistics.

Nitin Madnani, Jill Burstein, John Sabatini, and Tenaha OReilly. 2013. Automated scoring of a summary writing task designed to measure reading comprehension. *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–168.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence*, volume 1, pages 775–780.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

George A Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. 2007. The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing*, 4(2):4.

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. Association for Computational Linguistics.

Dragomir R. Radev, Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda Celebi, Stanko Dimitrov, Elliott Drabek, Ali Hakim, Wai Lam, and Danyu Liu. 2004. MEAD - A platform for multi-document multilingual text summarization. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomas Kovcisky, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*.

Vasile Rus and Mihai Lintean. 2012. A comparison of greedy and optimal assessment of natural language student input using word-to-word similarity metrics. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 157–162. Association for Computational Linguistics.

Stefan Ruseti, Mihai Dascalu, Amy M. Johnson, Danielle S. McNamara, Renu Balyan, Kathryn S. McCarthy, and Stefan Trausan-Matu. 2018. Scoring Summaries Using Recurrent Neural Networks. In *Intelligent Tutoring Systems*, pages 191–201, Cham. Springer International Publishing.

Craig Saunders, Alexander Gammerman, and Volodya Vovk. 1998. Ridge Regression Learning Algorithm in Dual Variables. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 515–521.

David Wade-Stein and Eileen Kintsch. 2004. Summary Street: Interactive computer support for writing. *Cognition and Instruction*, 22(3):333–362.

Shuohang Wang and Jing Jiang. 2016. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905*.

Cyril J. Weir, Ivana Vidakovic, and Evelina Galaczi. 2013. *Measured constructs*. Cambrdige University Press.

Denise Whitelock, Debora Field, Stephen Pulman, John T. E. Richardson, and Nicolas Van Labeke. 2013. OpenEssayist: an automated feedback system that supports university students as they write summative essays. In *Proceedings of the 1st International Conference on Open Learning: Role, Challenges and Aspirations*.

Menglin Xia, Ekaterina Kochmar, and Ted Briscoe. 2016. Text readability assessment for second language learners. In *Proceedings of the 11th Workshop on Building Educational Applications Using NLP*, pages 12–22. Association for Computational Linguistics.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 19–27.