Query-Focused Extractive Summarization based on Deep Learning: Comparison of Similarity Measures for Pseudo Ground Truth Generation

Yuliska † and Tetsuya SAKAI †

† Department of Computer Science and Engineering, Waseda University
3–4–1 Okubo, Shinjuku-ku, Tokyo 169–8555, Japan
E-mail: †yuliska0791@fuji.waseda.jp, †tetsuyasakai@acm.org

Abstract Query-focused summarization aims to produce a single, short document that summarizes a set of documents that are relevant to a given query. While some deep learning approaches have recently been applied to solve this task, how to automatically generate reliable ground truth labels for training remains an open problem. In this study, we employ eight existing textual similarity measures to generate ground truth labels at the sentence level given a reference summary. We then feed these different labelled data to deep learning approaches to generate extractive summaries. We use the DUC 2005-2007 benchmark datasets in our experiment. Our study shows that ROUGE-WE2 and ROUGE-SU measures achieved the best ROUGE scores in all deep neural models we employed.
Keywords Query-Focused Summarization, Extractive Summarization, Ground Truth Generation, Similarity Measures, Deep Neural Network

1 Introduction

Text summarization has recently gained much attention in natural language processing due to its promise in various applications. Examples include search engine snippets generation, news article headlines generation, question answering, and personalized recommendation engines. Among them, query-focused summarization (QFS), i.e, the task of producing a short and concise summary of a document or a set of documents based on user's query, remains a highly challenging task.

Recently, deep learning has been applied to text summarization and the majority of such approaches are extractive. Cao et al. [19] addressed the problem of query-focused extractive summarization using query-attention-weighted CNNs (Convolutioal Neural Network). Ren et al. [14] proposed Query Sentence Relation (QSR) which also used CNN with attention mechanism while Cheng and Lapatta [3] employed CNN and RNN (Recurrent Neural Network) as sentence extractor.

Neural-based Extractive summarization is often regarded as classification problem [19] [15] [16] [3]. Thus, ground truth labels are needed for training so that the model can generate correct prediction in the form of membership probability of each sentence in the final summary. However, Document Understanding Conference (DUC)(注1) dataset as benchmark dataset in text summarization only contains manual abstractive summaries as ground truth. To solve this problem, recent extractive summarization studies applied an unsupervised approach to convert the abstractive summaries to extractive labels. Nallapati et al. [15] and Nallapati and Ma [16] employed the following variants of ROUGE (Recall Oriented Study for Gisting Evaluation) [10]: ROUGE-1, ROUGE-2, and ROUGE-L similarity measures to generate ground truth in the form of sentence-level binary labels while Cao et al. [19] and Ren et al. [14] only utilized ROUGE-2. Cheng and Lapatta [3] used a rule-based system which is similar to ROUGE-1 and ROUGE-2 to form ground truth labels. These studies applied different similarity measures to generate ground truth, but which measure is the most reliable remains an open problem.

In this study, we focus on query-focused extractive summarization and reimplement eight textual similarity measures to generate ground truth at sentence level. We then feed these different labelled data to deep learning models to extract summaries and analyze their result. Our study shows that ROUGE-WE2 and ROUGE-SU measures achieved the best

(注1) : https://duc.nist.gov/data/

ROUGE scores in all deep neural models we employed.

2 Related Work

2.1 Query-Focused Extractive Summarization

Studies on query-focused extractive summarization spans a large range of approaches. Early studies on this task mostly used unsupervised graph-based approach to extract both salient and query-dependent sentences [11] [2], where nodes are sentences and the edge scores reflect the similarity between sentences, each node is given a relevance weight based on its relevance to the query. Following that, supervised machine learning approaches are applied to solve a query-focused summarization task. Ouyang et al. [18], Daume III and Marcu [7], Conroy et al. [8] used Support Vector Regression (SVR), Bayesian Statistical Model (BAYESUM), and Hidden Markov Model (HMM) respectively to extract query-dependent and query-independent features and thereby estimate the importance of sentences.

2.2 Extractive Summarization based on Deep Learning

Following the popularization of deep learning, many summarization systems have also employed deep learning techniques to address both general and query-focused summarization. Cheng and Lapatta [3] treated single document summarization as a sequence labelling task by utilizing CNN as sentence encoder and LSTM (Long Short-Term Memory) as sentence extractor. Cao et al. [19] addressed the problem of query-focused extractive summarization using query-attention-weighted CNNs. Ren et al. [14] proposed Query Sentence Relation (QSR) which also used CNN with attention mechanism. Kobayashi et al. [5] simply used the sum of trained word embeddings as sentence or document representation.

2.3 Ground Truth in Extractive Summarization

Many extractive summarization systems take a sentence classification approach. Hence, how to generate reliable ground truth in the form of sentence label is one of the primary issues [12]. Ouyang et al. [18] generated ground truth labels by using several N-gram based methods to compute similarity between sentence and reference or gold summary. Nallapati et al. [15] and Nallapati and Ma [16] employed ROUGE-1, ROUGE-2, and ROUGE-L similarity measures to generate ground truth in the form of sentence-level binary labels while Cao et al. [19] and Ren et al. [14] only utilized ROUGE-2. Cheng and Lapatta [3] used combination of unigram and bigram to form ground truth labels.

In the present study, we apply eight textual similarity measures to generate ground truth in the form of sentence-level binary labels for query-focused extractive summarization system based on deep learning. In our experiment, we study the effectiveness of each textual similarity in generating ground truth to identify the most reliable measure.

3 Approach

This section describes the DUC 2005-2007 datasets, textual similarity measures we employed, and the general scheme of ground truth generation. Following that, we explain about some deep learning models we utilized as extractive summarizer.

3.1 Dataset

Our experiments are conducted on the DUC 2005-2007 datasets. All the documents are from news articles and clustered into various thematic clusters. In DUC 2005, there are 4-9 reference summaries in each cluster. While in DUC 2006-2007, there are four reference summaries in each cluster. These reference summaries are created by NIST (National Institute Standards and Technology) assessors which consists of approximately 250 words. Table 1 shows the statistics of the three datasets. We use DUC 2005 as the training set, DUC 2006 as the evaluation set, and DUC 2007 as the test set.

Table 1 Statistics of DUC 2005-2007 Datasets

Year	Clusters	Sentences	Data Source
2005	50	45931	TREC
2006	50	34560	AQUAINT
2007	45	24282	AQUAINT

3.2 Similarity Measures

The following are the similarity measures that we apply for generating the ground truth:

3.2.1 ROUGE

We use the following ROUGE measures:

ROUGE-N: ROUGE-N is an N-grams recall between a system summary (sentence in pseudo ground truth generation) and a set of reference summaries. It is computed as follows:

$$R_{N\text{-}gram} = \frac{Count_{match}(Reference_{N\text{-}gram}, Summary_{N\text{-}gram})}{Count(Reference_{N\text{-}gram})}$$

$Count_{match}$ is the overlapping number of N-grams of reference summary and system summary and $Count$ is the total number of N-grams. Our goal is to give ROUGE score to each sentence, but the gap of the length between sentences and reference summaries are quite big, so it is not wise to score sentences based on ROUGE-N based recall only, because we want to also take the sentence length into consideration. To alleviate this problem, we use the ROUGE-N

based F-Measure, which is computed as follows:

$$P_{N\text{-}gram} = \frac{Count_{match}(Reference_{N\text{-}gram}, Summary_{N\text{-}gram})}{Count(Summary_{N\text{-}gram})}$$

$$F_{N\text{-}gram} = \frac{(1+\beta^2)R_{N\text{-}gram}P_{N\text{-}gram}}{R_{N\text{-}gram}+\beta^2 P_{N\text{-}gram}}$$

The ROUGE-N based F-Measure ($F_{N\text{-}gram}$) combines the recall and precision computation, so it will give a fair score to the long and short sentences. We set the value of β to 1 and we let N = 1,2 for ROUGE-1 and ROUGE-2 respectively.

ROUGE-L: ROUGE-L computes the Longest Common Subsequence (LCS) of a system summary and reference summaries. Let X and Y be a sentence and a reference summary respectively, given those two sequences, X and Y, the longest common sequence of X and Y is a common subsequence with maximum length. We use LCS based F-Measure, as follows:

$$R_{(lcs)} = \frac{LCS(X,Y)}{m}$$

$$P_{(lcs)} = \frac{LCS(X,Y)}{n}$$

$$F_{(lcs)} = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs}+\beta^2 P_{lcs}}$$

where m is the length of the reference summary and n is the length of the sentence.

ROUGE-SU: ROUGE-SU computes SKIP-Bigram between a system summary and a set of reference summaries, with the addition of unigram as counting unit. SKIP-Bigram is any pair of words in their sentence order, allowing for arbitrary gaps. We use SKIP-Bigram based F-Measure. It is computed as follows:

$$R_{(skip)} = \frac{SKIP\text{-}Bigram(X,Y)+Unigram}{m}$$

$$P_{(skip)} = \frac{SKIP\text{-}Bigram(X,Y)+Unigram}{n}$$

$$F_{(skip)} = \frac{(1+\beta^2)R_{skip}P_{skip}}{R_{skip}+\beta^2 P_{skip}}$$

ROUGE-SU comes with skip-distance parameter that defines the distance between the word during SKIP-Bigram creation. In this study, we set skip-distance parameter to 4.

ROUGE-Comb: A combination of ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-SU.

3.2.2 ROUGE-WE (Word Embeddings)

ROUGE is biased toward surface lexical similarities which makes it unsuitable for evaluating summaries with substantial paraphrasing. ROUGE-WE [13] comes with a solution to overcome this shortcoming. Instead of using N-grams overlap, it utilizes word embeddings to determine the similarity between words in a system summary and reference summaries. Thus, it gives a better evaluation for summarization task. The following explains how word embeddings can be incorporated into ROUGE.

Formally, ROUGE defines the following scenario to compute similarity between two words:

$$f_R(w_1, w_2) = \begin{cases} 1, & \text{if } w_1 = w_2 \\ 0, & \text{otherwise} \end{cases}$$

In ROUGE-WE, it defines similarity by the following:

$$f_{WE}(w_1, w_2) = \begin{cases} 0, & \text{if } v_1 \text{ or } v_2 \text{ are OOV} \\ v_1.v_2, & \text{otherwise} \end{cases}$$

OOV here means a situation where one of the words compared is not in the word embeddings vocabulary. There are 3 variants of ROUGE-WE in original paper: ROUGE-WE1, ROUGE-WE2 and ROUGE-WE-SU. We only employ ROUGE-WE2 and ROUGE-WE-SU in our experiment since these measures have good correlations with human judgements. For word embeddings, we use the same pretrained word embeddings that we use for training.

3.2.3 Keyword Overlap

We combine some variants of ROUGE (ROUGE-Comb) with the query keywords overlapping number. To extract keywords from query, we utilize the NLTK(注2) parser to generate parsing tree from each sentence. The words in the query are considered to be keywords if they are the tags: NN, NNS, NNP, NNPS, VB, VBD, VBG, VBN, VBP, and VBZ.

3.2.4 Embedding Similarity

Following the study of Kobayashi et al. [5], we use pretrained word2vec (same as that we use for training) to transform the words in the sentences into high dimensional vectors and get the average of those word vectors as sentence vectors. For reference summary representation, we apply a similar approach. To compute the similarity between sentence X and reference summary Y, we use cosine similarity as follows:

$$Cos(Y, X) = \frac{V(X)*V(Y)}{||V(X)|*|V(Y)||}$$

3.3 Pseudo Ground Truth Generation

Figure 1 explains how we generate the ground truth by utilizing textual similarity measures described in previous section. Given sequence of sentences $S = [s_1, s_2, \cdots, s_n]$ and sequence of abstractive summaries $M = [m_1, m_2, \cdots, m_n]$, similarity measures will give a score to each sentence based on its similarity to each abstractive summary. We then sort the sentences in descending order based on their similarity

(注2) : https://www.nltk.org/

score, and finally label the top K sentences as "1" and "0" otherwise. Regarding the value for K, we experimented with 5%-50% of the training set and decided to label the top 10% scored sentences as "1" as it derived the best result. This ground truth generation approach is based on the idea that the sentences with label "1" should be the one that maximize the ROUGE score with respect to gold summaries and highly relevant to the related query as gold summary also contains information which is questioned in query. Thus, the models should also give a high membership score to the sentence with label "1".

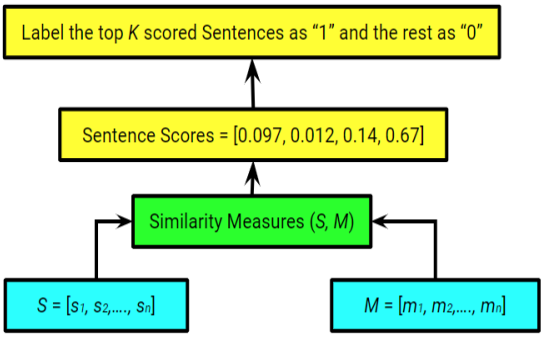

Figure 1 Ground Truth Generation Scheme

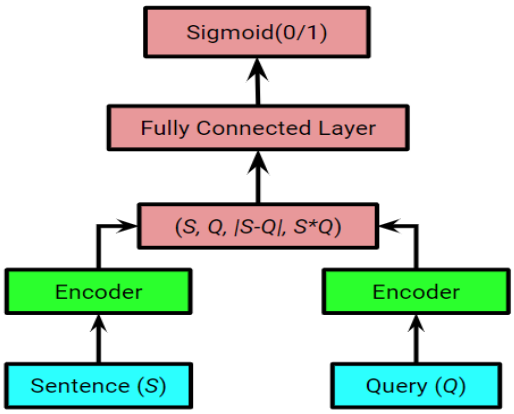

Figure 2 General Training Scheme

3.4 Deep Learning Models

This subsection introduces the models we considered for achieving query-focused extractive summarization. We prefer to treat extractive query-focused summarization as binary sentence classification task. We adopt the work of Conneau et al. [1] to train our model as depicted in Figure 2. As can be seen, we treat these models as an encoder which outputs sentence and query representations. After those representations are obtained, three operations are applied: (i) concatenation (S, Q); (ii) element-wise product $(S * Q)$; and (iii) absolute element-wise difference $(|S - Q|)$. The result vectors will contain the information of sentence and query relationship. These vectors are then fed into MLP to binary classify the sentence. As for encoders, we explain them as the following:

Bi-LSTM-Max: A concatenation of forward and backward LSTMs. Hence, the representation produced using this network combines information from forward and backward direction of sentence and question. We perform Max-pooling operation by selecting the maximum value over each dimension of the hidden units [4].

HieConv: We adopt this network architecture from Conneau et al. [1]. It consists of four layers of CNN where at every layer, a Max-pooling operation is applied over the feature maps. Then, the concatenation of each Max-pooling output is the final representation.

Stacked LSTM: Herman and Schrauwen [6] shows that stacking multiple Recurrent Neural Network (RNN) can potentially improve the sequence prediction problem. In our study, we attempt to stack two LSTM layers on top of each other, making the model capable of learning higher-level of representation both sentence and query.

3.5 Summary Sentences Selection

To select summary sentences, we employ a greedy approach as in many previous studies. Specifically, at test time, we sort the sentences in descending order according to the derived membership score from the model. We add a new sentence to the current summary if it contributes new bigrams to a certain degree. More specifically, if at least 50% of the bigrams from candidate sentences are new, we add it to the summary, until the summary contains approximately 250 words.

4 Experiment

4.1 Experimental Setup

We use NLTK to perform preprocessing as well as keyword extraction. All neural models are implemented on Keras(注3). The 300 dimensional pretrained Word2Vec(注4) vectors are used as word embeddings and the word embeddings are fine-tuned during training. We set the same hidden size for all models which is set to 50. For CNN, we apply ReLU activation function and set the filter to 2. For LSTM and Bi-LSTM, we apply Tanh activation. Pooling size is set to 2 at Max-pooling layer, for all models that is applied Max-pooling. Dropout operation is performed before feeding the word embeddings to the neural models and at every layer of network, we set the ratio of dropout to 0.5. In all neural networks, we also apply weight regularization (L2 norm) with the weight is set to 10^{-3}. The hidden size of MLP lay-

(注3) : https://keras.io/
(注4) : https://code.google.com/archive/p/word2vec/

Table 2 *Mean* ROUGE Scores (%) on Three Neural Models. Emb-Sim: Embedding Similarity, Comb+Query: Combination of ROUGE and keyword overlapping.

Model	Sim Measures	Evaluation Measures			
		ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU
Bi-LSTM-Max	ROUGE-2	41.00	10.51	18.10	15.99
	ROUGE-L	42.46	10.86	18.30	16.93
	ROUGE-SU	**43.56**	11.60	18.56	**17.73**
	Comb	43.33	11.43	18.63	17.63
	ROUGE-WE2	42.66	**11.93**	**19.02**	17.33
	ROUGE-WE-SU	41.84	11.26	17.97	16.61
	Emb-Sim	42.76	10.75	18.32	17.34
	Comb+Query	42.56	11.30	18.66	17.11
HieConv	ROUGE-2	41.16	10.37	17.99	16.03
	ROUGE-L	43.05	10.91	18.63	17.55
	ROUGE-SU	**43.26**	11.30	18.43	**17.57**
	Comb	42.89	11.07	18.46	17.35
	ROUGE-WE2	42.48	**11.75**	**18.79**	17.24
	ROUGE-WE-SU	42.27	11.24	18.37	16.99
	Emb-Sim	41.62	10.27	18.00	16.41
	Comb+Query	42.22	10.81	18.40	16.86
Stacked LSTM	ROUGE-2	40.63	10.23	17.98	15.93
	ROUGE-L	42.45	10.57	17.91	16.81
	ROUGE-SU	**42.94**	11.14	18.32	**17.43**
	Comb	42.19	10.84	18.02	16.63
	ROUGE-WE2	41.29	**11.40**	18.31	16.24
	ROUGE-WE-SU	41.96	11.04	**18.35**	16.85
	Emb-Sim	41.97	10.65	18.10	16.59
	Comb+Query	41.99	10.46	18.09	16.56

Table 3 Tukey HSD Test Results on Three Different Models. p: p-value, ES: Effect Size.

		ROUGE-L	ROUGE-SU	Comb	ROUGE-WE2	ROUGE-WE-SU	Comb+Query	Emb-Sim
Bi-LSTM-Max	ROUGE-2	p=0.9958	p=0.2377	p=0.3347	p=**0.0006**	p=0.3761	p=0.6205	p=0.9999
		ES= 1.2010	ES=0.9431	ES=1.1888	ES= 0.9330	ES=1.2179	ES=1.0983	ES=1.2955
	ROUGE-L	-	p=0.7011	p=0.8087	p=0.0640	p=0.8435	p=0.9637	p=0.9999
		-	ES=1.2777	ES=1.1936	ES=1.3986	ES=1.4592	ES=1.2543	ES=1.1855
	ROUGE-SU	-	-	p=0.9999	p=0.9040	p=0.9999	p=0.9986	p=0.5768
		-	-	ES=0.9887	ES=0.8452	ES=1.2290	ES=1.2543	ES=1.1112
	ROUGE-Comb	-	-	-	p=0.8247	p=1.000	p=0.9998	p=0.6988
		-	-	-	ES=0.9433	ES=1.2749	ES=1.3451	ES=12863
	ROUGE-WE2	-	-	-	-	p=0.7879	p=0.5521	p=**0.0377**
		-	-	-	-	ES=1.0238	ES=1.2095	ES=1.4246
	ROUGE-WE-SU	-	-	-	-	-	p=0.9999	p=0.7418
		-	-	-	-	-	ES=1.2516	ES=1.3208
	Comb+Query	-	-	-	-	-	-	p=0.9163
		-	-	-	-	-	-	ES=1.4372
HieConv	ROUGE-2	p=0.9875	p=0.4356	p=0.9386	p=0.0986	p=0.6079	p=0.9949	p=0.9999
		ES=1.2246	ES=1.0872	ES=1.2134	ES=1.0662	ES=1.0013	ES= 1.1311	ES=1.1264
	ROUGE-L	-	p=0.9343	p=0.9999	p=0.5388	p=0.9821	p=1.000	p=0.9667
		-	ES=0.8587	ES= 1.1255	ES=1.2870	ES=1.0748	ES=0.8379	ES=1.1700
	ROUGE-SU	-	-	p=0.9861	p=0.9960	p=0.9999	p=0.8941	p=0.3327
		-	-	ES=0.8645	ES=0.9654	ES=1.0241	ES=0.8379	ES=1.1256
	ROUGE-Comb	-	-	-	p=0.7341	p=0.9982	p=0.9999	p=0.8830
		-	-	-	ES=0.9849	ES=0.9946	ES=1.0312	ES=1.2835
	ROUGE-WE2	-	-	-	-	p=0.9767	p=0.4565	p=0.0636
		-	-	-	-	ES=1.0913	ES=1.0514	ES=1.1863
	ROUGE-WE-SU	-	-	-	-	-	p=0.9647	p=0.4946
		-	-	-	-	-	ES=1.0302	ES=1.3883
	Comb+Query	-	-	-	-	-	-	p=0.9834
		-	-	-	-	-	-	ES=1.1994
Stacked LSTM	ROUGE-2	p=0.9729	p=0.5565	p= 0.9080	p=0.1121	p=0.5183	p=0.9984	p=0.9383
		ES=1.3743	ES=1.2481	ES=1.1688	ES=0.9556	ES=1.1556	ES=1.0046	ES=0.9696
	ROUGE-L	-	p=0.9872	p=0.9999	p=0.6605	p=0.9818	p=0.9999	p= 0.9999
		-	ES=1.3993	ES=1.0583	ES=1.2222	ES=1.4417	ES=1.2374	ES=1.1360
	ROUGE-SU	-	-	p=0.9985	p=0.9895	p=1.000	p=0.9114	p=0.9964
		-	-	ES=1.1512	ES=1.3092	ES=1.2331	ES=1.2374	ES=1.0421
	ROUGE-Comb	-	-	-	p=0.8198	p=0.9975	p=0.9982	p=1.000
		-	-	-	ES=1.2670	ES=1.2327	ES=0.9343	ES=0.9321
	ROUGE-WE2	-	-	-	-	p=0.9930	p=0.4048	p=0.7658
		-	-	-	-	ES=1.2384	ES=1.0534	ES=1.0672
	ROUGE-WE-SU	-	-	-	-	-	p=0.8911	p=0.9943
		-	-	-	-	-	ES=1.7193	ES=1.2806
	Comb+Query	-	-	-	-	-	-	p=0.9993
		-	-	-	-	-	-	ES=0.9510

ers are 100, 50, and 1. We implement minibatch gradient descent using Adam [9] with learning rate is set to 10^{-2} and batch size = 100. To evaluate the quality of summary, we use variants of recall-based ROUGE metric for a deeper analysis: ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-SU. To compare the performance of each textual measure on each model statistically, we perform Tukey HSD (Honestly Significant Different) test based on two-way ANOVA (without replication) [17].

4. 2 Result and Discussions

Table 2 reports on ROUGE scores of the summaries obtained based on different kinds of textual similarity measures for pseudo ground truth generation on three deep neural models.

Table 3 shows the Tukey HSD p-values and effect sizes (ES) (i.e, standardized mean differences) based on two-way ANOVA (without replication) of different kinds of textual similarity measures for pseudo ground truth generation on three deep neural models. We follow the study of Cao et al. [19] which considered ROUGE-2 as the main evaluation score, hence, we perform the Tukey HSD test based on ROUGE-2 scores of output summaries.

In Bi-LSTM with Max-pooling model, ROUGE-SU outperforms the other similarity measures. It achieves the best ROUGE scores in general, with 43.56, 11.6, 18.56, and 17.73 for ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-SU respectively. However, the differences between ROUGE-SU and the other measures are not statistically significant (p-value > 0.05). ROUGE-WE2 comes as the second-best measure with 42.66, 11.93, 19.02, and 17.33 for ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-SU respectively. Nonetheless, ROUGE-WE2 only statistically significantly outperforms ROUGE-2 (p-value = 0.0006) and Emb-Sim (p-value = 0.0377). Meanwhile, we can easily notice that ROUGE-2 clearly underperforms the other similarity measures.

In HieConv model, ROUGE-WE2 measures achieves the best ROUGE scores with 42.48, 11.75, 18.79, and 17.24 for ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-SU respectively. But, its differences to the other similarity measures are not statistically significant. It is followed by ROUGE-SU measure that is outperformed by ROUGE-WE2 moderately with 43.26, 11.30, 18.43, and 17.57 for ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-SU respectively although it is not statistically better than the rest measures. In this model, ROUGE-2 measure also noticeably underperforms the other similarity measures.

In Stacked LSTM model, it is clear that ROUGE-SU measure outperforms the other similarity measures with 42.94, 11.14, 18.32, and 17.43 for ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-SU respectively. Nevertheless, it does not statistically outperform the other measures. ROUGE-WE2, ROUGE-Comb, and ROUGE-WE-SU have indistinguishable performances. Among them, ROUGE-WE-SU achieves the best ROUGE scores. This model also has a similar trend for ROUGE-2 measure which obtains the worst ROUGE scores among the measures compared.

Furthermore, we also notice a specific trend for ROUGE-SU and ROUGE-WE2 in all models. ROUGE-SU tends to generate summaries which have the best ROUGE-1 and ROUGE-SU scores among the measures compared. As for ROUGE-WE2, it tends to output summaries which have good ROUGE-2 and ROUGE-L scores among the measures compared.

5 Conclusion

Among the textual similarities that we explored in three deep neural networks, ROUGE-WE2 and ROUGE-SU achieved the best ROUGE scores. However, from Table 3, it seems that ROUGE-WE2 only statistically significantly outperformed ROUGE2 and Emb-Sim measures for Bi-LSTM with Max-pooling model, but otherwise the differences between different measures were *not* statistically significant. Overall, we suggest that ROUGE-WE2 and ROUGE-SU are the best measures among the textual similarity measures compared and ROUGE-2 might be the worst measure for pseudo ground generation.

References

[1] Alexis Conneau, Douwe Kiela, H. S. L. B. and Bordes, A.: Supervised Learning of Universal Sentence Representations from Natural Language Inference Data, *Proceeding of the 2017 Conference on Empirical Methods on Natural Language Processing*, pp. 670–680 (2017).

[2] Bhaskar, P. and Bandyopadhyay, S.: A Query Focused Multi-Document Automatic Summarization, *Proceeding of Twenty-Four Pacific Asia Conference on Language, Information and Computation*, pp. 545–554 (2010).

[3] Cheng, J. and Lapatta, M.: Neural Summarization by Extracting Sentences and Words, *Proceeding of 54th Annual Meeting of the Association for Computational Linguistics*, pp. 484–494 (2016).

[4] Collobert, R. and Weston, J.: A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning, *Proceeding of the 25th International Conference on Machine Learning*, pp. 160–167 (2008).

[5] Hayato Kobayashi, M. N. and Yatsuka, T.: Summarization Based on Embedding Distributions, *Proceeding of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1984–1989 (2015).

[6] Hermans, M. and Schrauwen, B.: Training and Analyzing Deep Recurrent Neural Network, *In Advances in Neural Information Processing System 26*, pp. 190–198 (2013).

[7] III, H. D. and Marcu, D.: Bayesian Query-Focused Summarization, *Proceeding of the 21st International Conference on*

Computational Linguistics and 44th Annual Meeting of the ACL, pp. 305–312 (2006).

[8] Jhon M. Conroy, J. D. S. and Stewart, J. G.: Classy Query-Based Multi-Document Summarization, *Proceeding of the Document Understanding Conf. Wksp. 2005 at the Human Language Technology Conf./Conf. on Emprical Methods in Natural Language Processing (HLT/EMNLP)* (2006).

[9] Kingma, D. P. and Ba, J. L.: Adam: A Method for Stochastic Optimization, *Proceeding of the 3rd International Conference on Learning Representations* (2015).

[10] Lin, C.-Y.: ROUGE: A Package for Automatic Evaluation of Summaries, *Proceedings of the ACL-04 Workshop*, pp. 74–81 (2004).

[11] M. Sravanti, C. R. C. and Kumar, P. S.: QueSTS: A Query Specific Summarization System, *Proceeding of the Twenty-First International FLAIRS Conference*, pp. 219–224 (2008).

[12] Mehdi Allahyahri, Seyedamin Pouriyeh, M. A. S. S. E. D. T. J. B. G. and Kochut, K.: Text Summarization: A Brief Survey, *International Journal of Advanced Computer Science and Applications, Vol. 8, No. 10*, pp. 397–405 (2017).

[13] Ng, J.-P. and Abrecht, V.: Better Summarization Evaluation with Word Embeddings for ROUGE, *Proceeding of the 2015 Conference on Empirical Methods on Natural Language Processing*, pp. 1925–1930 (2015).

[14] Pengjie Ren, Zhumin Chen, Z. R. F. W. L. N. J. M. and Ridjke, M. D.: Sentence Relation for Extractive Summarization with Deep Neural Network, *ACM Transaction on Information System (TOIS), Volume 36 Issue 4, Article No. 39* (2018).

[15] Ramesh Nallapati, B. Z. and Ma, M.: Classify or Select: Neural Architectures for Extractive Document Summarization, *Under Review as Conference Paper at ICLR 2017* (2017).

[16] Ramesh Nallapati, F. Z. and Zhou, B.: SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents, *Proceeding of Thirty-First AAAI Conference on Artificial Intelligence*, pp. 3075–3081 (2017).

[17] Sakai, T.: Laboratory Experiments in Information Retrieval: Sample Sizes, Effect Sizes, and Statistical Power, *Springer*, pp. 51–80 (2018).

[18] You Ouyang, Wenjie Li, S. L. and Lu, Q.: Applying Regression Model to Query-Focused Multi-Document Summarization, *International Journal of Information Processing and Management*, pp. 227–237 (2011).

[19] Ziqiang Cao, Wenjie Li, S. L. F. W. and Li, Y.: Attsum: Join Learning of Focusing and Summarization with Neural Attention, *Proceeding of COLING 2016, the 26th International Conference on Computational Linguistic: Technical Paper*, pp. 547–556 (2017).