University of Houston @ CL-SciSumm 2018[⋆]

Luis F.T. De Moraes[1], Avisha Das[1], Samaneh Karimi[1,2], and Rakesh Verma[1]

[1] Computer Science Department
University of Houston, Houston, TX 77204, USA
ltdemoraes@uh.edu, adas5@uh.edu, rverma@uh.edu
[2] University of Tehran, Iran
samanekarimi@ut.ac.ir

Abstract. In this paper we present our methods and their results on the CL-SciSumm tasks of 2018. In this round, for Task 1A, we tried deep learning methods, a variation of the Positional Language Model and also our methods from BIRNDL 2017. The results show that the deep learning method outperforms the positional language model method and TFIDF method from BRINDL 2017 yields the best F1 score. For Task 1B, we used rule-based methods and classifiers.

1 Introduction

Constructing summaries of scientific papers is useful for combating the exponential growth of scientific research. Automatic summarization of news articles is a well-studied problem [1, 3, 17], but scientific paper summarization has been relatively less studied. In [1], researchers showed that, for scientific papers, it is possible to beat the baselines in a statistically significant way. On the other hand, for news articles, this is quite difficult to achieve.

The CL-SciSumm series of shared tasks has been organized to give a boost to summarization of scientific research. The emphasis of these tasks is to construct a summary of a scientific paper based on the citations of the paper. The idea is that the citations of the paper represent the impact it has had and could therefore be used to generate a potentially more interesting and useful summary.

The CL-SciSumm series has three tasks [6]. In Task 1A, given a scientific paper P (called "reference document") and a citance c of P, the goal is to retrieve the most relevant sentences from P considering c as a query. These sentences are called the *reference span* of c, In Task 1B, the goal is to classify the reference span into one of five-predefined categories: method, aim, etc. In Task 2, the goal is to construct a summary of P based on the reference spans corresponding to all the citances of P.

This year we participated in the first two tasks, Tasks 1A and 1B. Our methods for Task 1A were: a sentence similarity method using Siamese Deep Learning Networks [16] and a Positional Language Model approach [12]. For

[⋆] Research supported in part by NSF grants CNS 1319212, DGE 1433817, DUE 1356705 and DUE 1241772

Task 1B, we have the same method we employed last year, which includes a *Rule-based method* augmented by WordNet expansion; a *Machine learning based method* using four classifiers: SVMs, Random Forests, Decision Trees, and Multilayer Perceptron; and an ensemble method: AdaBoost. TF-IDF features are used to train all classifiers.

2 Related Work

The CL-SciSumm series of tasks has led to several submissions by researchers from all over the world, and follow-ups and works by other researchers. We briefly summarize the most directly related works here and refer the reader to [11, 15, 8] for other related works. Most of the baseline algorithms used in this paper for Task 1A have been described in [11] and [8].

To our knowledge, no one else has tried positional language models for Task 1A. Positional Language Modeling along with textual entailment and Structural Correspondence Learning were used for reference span extraction by [8]. [13] used similarity measures like LDA similarity, TF-IDF similarity along with position based features for reference span extraction; while [2] used a query based approach where each citance can be used to extract related reference spans from the text. For more details, we refer the authors to [5].

Deep learning for citance-based summarization has been tried very recently in [7]. The researchers extracted and combine several classes of features like similarity based lexical measures from reference sentences as well as citances (word overlap, ROUGE measures, TF-IDF Similarity, etc.) and Word2Vec and WordNet-based similarity attributes. The features also included surface level features such as count of words, characters and numbers extracted from reference sentences. The feature engineering process was used to train two ensemble boosting algorithm based classifiers and a convolutional neural network (CNN). Their experiments combine datasets across the CL-SciSumm competitions [6] and they report good results with the CNN algorithm. Siamese Networks have been commonly used for detecting similarity between short pairs of sentences [16].

3 Dataset

The dataset is available as part of the CL-SciSumm 2018 Shared Task.[3] The training corpus consists of 40 scientific papers from the computational linguistics field, and the test set consists of 10 papers from the same domain.

4 Task 1A Methods

In order to find the most relevant sentences of a reference document corresponding to a citance – the main goal of Task 1A [6] – different approaches, such as

[3] https://github.com/WING-NUS/scisumm-corpus

machine learning, information retrieval, etc., can be employed. We have used several different methods and a couple of baselines for this task. We describe these methods and our preliminary results below.

4.1 Baselines

Our first baseline is TF-IDF. We first convert each sentence into a boolean unigram and bigram vector. If an n-gram is present, then its dimension contains the value 1; if the n-gram is absent, the value 0. We then reweigh the value of each n-gram according to its document-wide TF-IDF score, where IDF is calculated considering each sentence as a document. N-grams that appear in few sentences have greater weight because they are better at distinguishing between sentences. N-grams that appear in many sentences have lesser weight because they do not help us narrow down the sentence candidates as much. We then calculate the cosine similarity between the citance's vector and that of each sentence.

Our second baseline is word embeddings [14]. We use embeddings trained on the ACL Anthology.[4] For every sentence we convert each word into its corresponding embedding vector. Note that each sentence is now a bag of word embeddings. To determine the similarity between two sentences, we use the Word Mover's Distance [10] (also known as optimal transport). This metric essentially looks for the minimal distance necessary to move all vectors such that, in the end, every vector from one sentence overlaps a vector from the other sentence.

No matter which baseline, if a citance is composed of more than one sentence, we regard it as a single, long sentence. Since each baseline scores every sentence, we sort them and pick the top 3 sentences as our choice of reference span. For a discussion of the preprocessing steps we undertake and further details, we refer the reader to [17].

4.2 Deep Learning with Siamese Networks

We use a Siamese network-based framework [16] to model the semantic similarity between the citance and reference span. A Siamese network architecture consists of two or more sub-networks which together are used to learn the underlying semantic similarity between a given pair of sentences. For the purpose of Task 1A, we use a Siamese Network to model the nature of semantic similarity that exists between a citance and its reference span. Such a network, also tries to capture the dissimilarity between a citance and unrelated or irrelevant reference sentences from the given reference document.

For the architecture, we use Long Short Term Memory (LSTM) [4] Units to learn the dependencies across the textual pairs. The set of sentences extracted from the reference document have been filtered to only sentences of length 15 to 70 words. We also preprocess the reference sentences to remove non-ASCII characters and text between parentheses. The citance and reference pairs are converted to word embeddings using Word2Vec model on GoogleNews pre-trained

[4] http://aclweb.org/anthology/

word embeddings. For generating a pair from the annotated data provided, for every citance we select the annotated reference span. If a citance has more than one reference sentence in the span, we create separate pairs for each sentence with the citance.

We observed that assigning rigid labels of 0s and 1s to citance and reference sentence pairs lead to multiple misclassified instances. Therefore, instead of binary labels, we assign cosine similarity values as normalized scores ranging between 0 to 1 – for a given citance, the extracted reference spans from the annotated files are assigned a 1. For other citance-reference sentence pairs, we assign the similarity score. The Siamese network thus behaves as a regression model as it identifies semantically similar pairs of citance and reference sentences.

The proposed architecture has two bidirectional LSTM sub-networks with 50 hidden units each. We also add a dropout layer and densely connected output layer to each sub-network. We use the mean square error for calculating the training loss along with the AdaDelta [18] optimization function. The final similarity measure is the Manhattan similarity measure given by the following equation:

$$Ma\_Sim = exp(-||Out_{left} - Out_{right}||_1)$$

The LSTM network acts as an encoder to generate the semantic meaning between the given citance and reference sentence pair. The exponent of the negative of the absolute distance between the encoded LSTM outputs is used to calculate the Manhattan similarity between the sentences of the given pair. The Siamese architecture implemented for this Task iA based on the system described in [16].

The proposed Siamese Networks are trained on 35 documents from the training documents provided by the organizers[5]. We select the remaining 5 documents as our validation data set. In Table 1a, we present the precision, recall and F1-score values observed using two models trained for 1 epoch and 5 epochs respectively.

4.3 Positional Language Model

Positional language model is one of our approaches to solve the problem of Task 1A. In this approach, we transform the problem to identification of the best position in the reference document which relates to the citance based on the positional distribution of words in the reference document, i.e. PLM (positional language model). The PLM approach [12] utilizes the proximity information of words in documents to retrieve better results in response to query. For problem 1A, reference documents are considered as documents and citances as queries. In the PLM approach, a separate language model is constructed for each position of words in the document. The PLM of document d at position i is estimated as follows:

[5] https://github.com/WING-NUS/scisumm-corpus

$$p(w|d,i) = \frac{c^{'}(w,i)}{\sum_{w' \epsilon V} c^{'}(w',i)},$$

wherein V denotes the vocabulary and $c^{'}(w,i)$ is the propagated count of word w at position i from all of its occurrences in the document.

The PLM approach is based on the assumption that the occurrence of each word at each position of the document can be propagated to other positions within the same document using a density function. The density function assigns higher propagation weights to terms that are closer to the position in the PLM. By having the PLMs for all of the positions in the document, a position-specific retrieval score can be computed for each position in the document in response to the query. This position-specific retrieval score is obtained by computing the similarity between the language model of the query and the PLM of that position using KL-divergence formula [9]. Therefore, if a citance includes more than one sentence, all of the citance's sentences impact the language model of the query simultaneously.

The position-specific retrieval scores can be used to compute an overall retrieval score for the document through different strategies. For instance, using best position strategy, the final retrieval score of the document is the score of its best matching position. In PLM method for task 1A, in order to find top N retrieval results from reference documents in response to each citance as query, two approaches are used: in the first approach, N most relevant sentences of the reference document that have highest retrieval scores based on their PLMs are returned as results. In the second approach, the best position in the reference document is selected as the top result and the rest of N results are chosen from its adjacent sentences. For both approaches, the PLM implementation released by the authors of [12] is used.

5 Task 1B Methods

For Task 1B, we applied our previous methods proposed in [8] on the 2018 datasets. The methods include a rule-based method, which is basically a comparison-based method augmented by WordNet expansion and a classification method. More details are available in [8].

6 Evaluation

The results of all four methods for Task 1A on training set 2018 are reported in Table 1a. We observe that the variation of PLM reported in this paper outperforms our previous variants of PLM reported in [8]. Both baselines outperform the deep learning and PLM approaches, which is a bit surprising. Perhaps, more feature engineering is required for these tasks. Although the baselines fare better in terms of F1 score, the Siamese Networks have a substantial advantage in terms of recall.

Method	Prec	Recall	F1
PLM	5.33	17.32	8.16
PLM-FWBW	3.66	11.89	5.60
Siamese-1E	5.00	71.00	9.34
Siamese-5E	4.50	35.00	7.97
TF-IDF	10.09	19.65	13.33
WordEmbed	10.00	19.48	13.22

(a) Scores (%) for Task 1A.

Method	Prec	Recall	F1
Rule_based-V1	32.80	30.53	31.62
Rule_based-V2	60.55	56.36	58.38
Rule_based-V3	69.98	65.14	67.47
Method_only	74.90	69.71	72.21
SVM	72.60	67.51	69.93
Random Forest	65.68	61.79	63.62
Decision Tree	50.70	54.08	52.23
MLP	63.09	59.44	61.17
Adaboost	52.72	54.73	53.61

(b) Scores (%) for Task 1B.

Table 1: Results on Training Set 2018

The results of Task 1B methods on training set 2018 are reported in Table 1b. For the classification methods, 10-fold cross-validated results are reported, the value of C in SVM is set to 0.02 and the MLP classifier used consists of three layers with 100, 50 and 20 nodes in the first, second and third layers, respectively.

7 Conclusion and Future Work

In this paper, we have presented our methods and preliminary results for Tasks 1A and 1B of the CL-SciSumm 2018 shared task. A lot more work can be done in terms of feature engineering. In addition, figuring out why some methods favor recall to the detriment of precision could help us in forming stronger ensembles.

References

1. Barrera, A., Verma, R.: Combining syntax and semantics for automatic extractive single-document summarization. In: CICLING. vol. LNCS 7182, pp. 366–377 (2012)

2. Felber, T., Kern, R.: Graz university of technology at cl-scisumm 2017: Query generation strategies. In: Proc. of the 2nd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2017). Tokyo, Japan (August 2017) (2017)
3. Gambhir, M., Gupta, V.: Recent automatic text summarization techniques: a survey. Artif. Intell. Rev. **47**(1), 1–66 (2017)
4. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
5. Jaidka, K., Chandrasekaran, M., Jain, D., Kan, M.Y.: The cl-scisumm shared task 2017: Results and key insights. In: Proceedings of the Computational Linguistics Scientific Summarization Shared Task (CL-SciSumm 2017), organized as a part of the 2nd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2017) (2017)
6. Jaidka, K., Chandrasekaran, M.K., Rustagi, S., Kan, M.Y.: Insights from CL-SciSumm 2016: the faceted scientific document summarization Shared Task. International Journal on Digital Libraries (jun 2017). https://doi.org/10.1007/s00799-017-0221-y, https://doi.org/10.1007/s00799-017-0221-y
7. Jha, S., Chaurasia, A., Sudhakar, A., Singh, A.K.: Reference scope identification for citances using convolutional neural network (2017)
8. Karimi, S., Moraes, L., Das, A., Verma, R.: University of Houston@ cl-scisumm 2017: Positional language models, structural correspondence learning and textual entailment. In: Proc. of the 2nd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2017). Tokyo, Japan (August 2017) (2017)
9. Kullback, S., Leibler, R.A.: On information and sufficiency. The annals of mathematical statistics **22**(1), 79–86 (1951)
10. Kusner, M.J., Sun, Y., Kolkin, N.I., Weinberger, K.Q., et al.: From word embeddings to document distances. In: ICML. vol. 15, pp. 957–966 (2015)
11. Lu, K., Mao, J., Li, G., Xu, J.: Recognizing reference spans and classifying their discourse facets. In: Proc. of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2016) (2016)
12. Lv, Y., Zhai, C.: Positional language models for information retrieval. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval SIGIR 09 (2009)
13. Ma, S., Xu, J., Wang, J., Zhang, C.: Njust@ clscisumm-17. In: Proc. of the 2nd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2017). Tokyo, Japan (August 2017) (2017)
14. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR **abs/1301.3781** (2013), http://arxiv.org/abs/1301.3781
15. Moraes, L.F.T., Baki, S., Verma, R.M., Lee, D.: University of Houston at cl-scisumm 2016: Svms with tree kernels and sentence similarity. In: Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL) co-located with the Joint Conference on Digital Libraries 2016 (JCDL 2016), Newark, NJ, USA, June 23, 2016. pp. 113–121 (2016), http://ceur-ws.org/Vol-1610/paper13.pdf
16. Mueller, J., Thyagarajan, A.: Siamese recurrent architectures for learning sentence similarity. In: AAAI. pp. 2786–2792 (2016)

17. Verma, R.M., Lee, D.: Extractive summarization: Limits, compression, generalized model and heuristics. Computacion y Sistemas **21**(4), 787–798 (2017)
18. Zeiler, M.D.: ADADELTA: an adaptive learning rate method. CoRR **abs/1212.5701** (2012), http://arxiv.org/abs/1212.5701