LaSTUS-TALN+INCO @ CL-SciSumm 2019

Luis Chiruzzo$^{1[0000-0002-1697-4614]}$, Ahmed AbuRa'ed$^{2[0000-0002-6241-7755]}$, Alex Bravo$^{2[1111-2222-3333-4444]}$, and Horacio Saggion$^{2[0000-0003-0016-7807]}$

[1] Universidad de la República, Facultad de Ingeniería, INCO, Montevideo, Uruguay
luischir@fing.edu.uy

[2] Universitat Pompeu Fabra, DTIC, LaSTUS-TALN, C/Tanger 122, Barcelona (08018), Spain
first.last@upf.edu

Abstract. In this paper we present several systems developed to participate in the 4th Computational Linguistics Scientific Document Summarization Shared challenge which addresses the problem of summarizing a scientific paper using information from its citation network (i.e., the papers that cite the given paper). Given a cluster of scientific documents where one is a reference paper (RP) and the remaining documents are papers citing the reference, two tasks are proposed: (i) to identify which sentences in the reference paper are being cited and why they are cited, and (ii) to produce a citation-based summary of the reference paper using the information in the cluster. Our systems are based on both supervised (LSTM and convolutional neural networks) and unsupervised techniques using word embedding representations and features computed from the linguistic and semantic analysis of the documents.

Keywords: Citation-based Summarization · Scientific Document Analysis · Convolutional Neural Networks · Text-similarity Measures.

1 Introduction

Although scientific summarization has always been an important research topic in the area of natural language processing (NLP) [13, 19, 24, 25] in recent years new summarization approaches have emerged which take advantage of the citations that a scientific article has received in order to extract and summarize its main contributions [20, 21, 1].

The interest in the area has motivated the development of a series of evaluation exercises in scientific summarization in the Computational Linguistics (CL) domain known as the Computational Linguistics Scientific Document Summarization Shared Task which started in 2014 as a pilot [9] and which is now a well developed challenge in its fourth year [7, 8].

In this challenge, given a cluster of n documents where one is a reference paper (RP) and the $n-1$ remaining documents are papers (i.e., citing papers (CPs)) citing the reference paper, participants of the challenge have to develop automatic procedures to simulate the following tasks: given a cluster of n documents where one is a reference paper and the $n-1$ remaining documents are papers containing citations to it:

The challange has the following tasks:

- **Task 1A**: For each citance in the citing papers (i.e., text spans containing a citation), identify the cited spans of text in the reference paper that most accurately reflect the citance.
- **Task 1B**: For each cited text span, identify which **discourse facet** it belongs to, among: *Aim*, *Hypothesis*, *Implication*, *Results*, or *Method*.
- **Task 2**: Finally, an optional task consists on generating a structured summary of the reference paper with up to 250 words from the cited text spans.

In this paper we report the systems developed at LaSTUS-TALN+INCO to participate in CL-SciSumm 2019 [6]. We include a supervised system based on recurrent neural networks and an unsupervised system based on sentence similarity for Task 1A, one supervised approach for Task 1B, and one supervised approach for Task 2. Except for the recurrent neural network method, the rest of the systems for Tasks 1A and 1B follow similar approaches to the ones reported in [4] and [2], achieving good performance in previous editions of the task. The approach for Task 2 follows the method described in [2] which, according to official results [10] [14], was the winning approach in CL-SciSumm 2018.

2 Task 1

We tried a supervised and an unsupervised approach for Task 1A. We separated the CL SciSumm 2018 corpus of documents in 75% for training and 25% for development evaluation. We also used the 978 documents from ScisummNet 2019 automatically annotated following [18] for pre-training our neural network models.

2.1 Supervised approach

Our supervised approach consists in a neural network architecture for finding out which sentences from the reference document are most the likely candidates for being referenced by a given citation.

Network architecture The neural networks have the following structure:

- **Input layer -** Two sentences: the citation text and a sentence from the reference document.
- **Embeddings layer -** We tried with two collections of embeddings: Google News[3] 300 dimensions vectors and BabelNet[5][16] 300 dimensions vectors.
- **LSTM layers -** One, two or three stacked bidirectional LSTM layers.
- **Dense layer -** One fully connected layer.
- **Output layer -** One unit indicating the probability that the sentence from the reference document corresponds to the citation.

[3] https://code.google.com/archive/p/word2vec/

We carried different experiments using word embeddings or BabelNet synset embeddings, the tokens in the input layer were words or synsets depending on the experiment. The LSTM layers combine up to three layers and a dense layer with sizes 150, 300, or 450. In all of our experiments we aimed to optimize against our development set, which contains 25% of the CL-SciSumm 2018 training set.

Pre-training and Training We separated the training of the models in two stages: pre-training and training. The 978 clusters of documents from the Yale corpus were used to do a pre-training of the LSTM models. During pre-training, we trained the models using 70% of the Yale corpus optimizing against the remaining 30% using early stopping.

After this pre-training phase was over, we trained the resulting model using our CL-SciSumm 2018 training partition. We found out that, in general, pre-training with the Yale corpus and then training with CL-SciSumm 2019 achieved better results than only training with CL-SciSumm, even if the Yale data was automatically annotated. For the training stage, we used early stopping optimizing against 20% of our training corpus.

2.2 Unsupervised approach

As in previous editions [2][4][3], we used an unsupervised approach consisting in comparing all the sentences in a reference document with a citation and returning the most similar one according to certain metric. In this case, we transformed all sentences and citations into BabelNet synsets and we took the centroid of the synsets as a way of creating a sentence embedding. Then we used cosine similarity two find out which of the candidate sentences were more suitable.

2.3 Voting System

We submitted a voting system which considers sentences picked by two or more of the previous mentioned systems for Task 1.

2.4 Development results

Table 1 shows the results over the development corpus for the different experiments we tried. In general, the neural networks performed worse over the development corpus than the simpler unsupervised method. The networks trained using Google News vectors achieved better results than the ones trained using BabelNet vectors. Notice that in each case the number of sentences in order to get the best results for development was different.

3 Task 2

In this section, we describe our extractive text summarization approach based on convolutional neural networks which extends on our previous work on trainable

Model	Layers	Size	Top n	Precision	Recall	F-1
Babelnet Cosine	-	-	5	6.08%	21.33%	9.46%
Google News	1	150	2	6.52%	8.86%	7.51%
Google News	2	150	3	5.80%	11.81%	7.78%
Google News	3	150	3	4.50%	12.24%	6.58%
Google News	1	300	2	5.59%	7.59%	6.44%
Google News	2	300	2	6.21%	8.44%	7.16%
BabelNet	1	150	9	2.67%	16.89%	4.61%
BabelNet	3	150	4	3.48%	9.78&	5.13%
BabelNet	1	300	20	1.39%	19.56%	2.60%
BabelNet	2	300	6	2.42%	10.22%	3.92%
BabelNet	3	300	10	1.96%	13.78%	3.43%

Table 1. Results for Task 1a over the development set.

summarization [23, 4]. The network generates a summary by selecting the most relevant sentences from the RP using linguistic and semantic features from RP and CPs. The aim of our CNN is to learn the relation between a sentence and a scoring value indicating its relevance.

3.1 Context Features

In order to extract the linguistic information from both sources (RP and CPs), we developed a complex feature extraction method to characterize each sentence in the RP and its relation with the corresponding CPs.

We extracted a set of numeric features some of which are based on comparing a sentence to its (document or cluster) context:

- Sentence Abstract Similarity Scores: the similarity of a sentence vector to the author abstract vectors (three features).
- Sentence Centroid Similarity Scores: the similarity of a sentence vector to the article centroid (three features).
- First Sentence Similarity Scores: the similarity of a sentence vector to the vector of the first sentence, that is, the title of the RP (three features).
- Position Score: a score representing the position of the sentence in the article. Sentences at the beginning of the article have high scores and sentence at the end of the article have low scores.
- Position in Section Score: a score representing the position of the sentence in the section of the article. Sentences in first section get higher scores, sentences in last section get low scores.
- Position in a Specific Section Score: a score representing the position of the sentence in a particular section. Sentences at the beginning of the section get higher scores and sentences at the end of the section get lower scores.
- TextRank Normalized Scores: a sentence vector is computed to obtain a normalized score using the TextRank algorithm [15] (three features).

- Term Frequency Score: we sum up the tf*idf values of all words in the sentence. Then, the obtained value is normalized using the set of scores from the whole article.
- Citation Marker Score: the ratio of the number of citation markers in the sentence to the total number of citation markers in the article.
- Rhetorical Class Probability Scores: probability of a sentence being in one of five possible rhetorical categories calculated by the Dr. Inventor framework [22].
- Citing Paper Maximum Similarity Scores: each RP sentence vector is compared to each citation vector in each CP to get the maximum possible cosine similarity (three features).
- Citing Paper Minimum Similarity Scores: each RP sentence vector is compared to each citation vector in each CP to get the minimum possible cosine similarity (three features).
- Citing Paper Average Similarity Scores: each RP sentence vector is compared to each citation vector and the average cosine value obtained (three features).

3.2 Scoring Values

As commented above, our CNN learns the relation between features and a score, that is, a regression task by devising various scoring functions to represent the likelihood of a sentence belonging to a summary (for abstract, community and human). The nomenclature followed to symbolize a scoring function is SC_{Sum}, where SC is the specific scoring function (which is indicated bellow) and Sum is any summary type: abstract (Abs), community (Com) or human (Hum). The scoring functions are defined bellow:

- Cosine Distance: we calculated the maximum cosine similarity between each sentence vector in the RP with each vector in the gold standard summaries. This method produced three scoring functions (SUMMA (SU_{Sum}), ACL (ACL_{Sum}), and Google (Go_{Sum})) for each summary type.
- ROUGE-2 Similarity: we also calculated similarities based on the overlap of bigrams between sentences in the RP and gold standard summaries. In this regard, each sentence in the RP is compared with each gold standard summary using ROUGE-2 [12]. The precision value from this comparison is taken for the scoring function and is symbolized as $R2_{Sum}$.
- Scoring Functions Average: Moreover, we computed the average between all scoring functions (SUMMA, ACL, Google and ROUGE-2) for each summary type. In addition, we also calculated a simplified average with vectors do not based on word-frequencies (ACL, Google and ROUGE-2). These scoring functions are indicated as Av_{Sum} and SAv_{Sum}, respectively.

Finally, these computation produced eighteen different functions to learn: SUMMA (SU), ACL (ACL) and Google (Go) vectors, ROUGE-2 ($R2$), Average (Av) and Simplified Average (SAv) times abstract (Abs), community (Com), human (Hum) summaries.

3.3 Convolution Model

Regarding the neural network hyperparameters, the CNN was defined with the Adadelta updater [26] and the gradients were computed using back-propagation as Kim [11] and Nguyen [17]. Also we used the sigmoid activation function, a dropout rate of 0.5, l2 constraint of 3. For the convolutions, we applied 3 filter window sizes (3, 4 and 5) to context features and 4 filter window sizes (2, 3, 4 and 5) to word embeddings. For each window were applied 150 filters for convolution. Finally, for learning the regression task we applied a Mean Squared Error (MSE) as loss function.

4 Challenge Submissions

For task 1, we sent the following four submissions:

- **run1:** LSTM trained with Babelnet vectors with three layers of size 150.
- **run2:** BabelNet centroids cosine similarity.
- **run3:** LSTM trained with Google News vectors with two layers of size 150.
- **run4:** Voting scheme based on [2].

For task 2, the submissions we sent are the following:

- Similarity with the abstract from all similarity scores except SUMMA.
- Similarity with the abstract from all scores.
- Rouge based score similarity with the abstract.
- ACL cosine similarity based score with the abstract.

Finally, based on [2] we presented the results of a classifier that addresses Task 1B of identifying the discourse facet for each identified cited sentence.

5 Results

The performance of our systems for task 1 over the test set is shown in table 2. We can see that the LSTM approached underperformed compared to their results over the development corpus, one possible cause for this is that the systems could have overfit to the training and development data. Out of the methods we tried, the system that performs best for task 1 is still the voting scheme based on [2]. The performance of our systems for task 2 over the test set is shown in table 3.

6 Conclusion

We have described the systems developed to participate in Tasks 1a, 1b and 2 in the CL-SciSumm 2019 summarization challenge. For Task 1a – which aimed at identifying cited sentences –, we implemented supervised and unsupervised methods. Our supervised systems are based on LSTM neural networks, while the

Run	Task1A		Task1B
	Sentence Overlap (F1)	ROUGE-SU4 (F1)	(F1)
run4 Voting scheme	0.070	0.025	0.122
run2 BabelNet centroids	0.066	0.026	0.277
run3 Google News LSTM	0.031	0.021	0.078
run1 BabelNet LSTM	0.020	0.015	0.070

Table 2. Test results for task 1.

Run	Abstract		Community		Human	
	R-2	R-SU4	R-2	R-SU4	R-2	R-SU4
run1	0.329	0.172	0.149	0.090	0.241	0.171
run2	0.316	0.167	0.169	0.101	0.245	0.169
run3	0.311	0.156	0.153	0.093	0.252	0.170
run4	0.246	0.147	0.131	0.084	0.170	0.141

Table 3. Test results for task 2.

unsupervised techniques take advantage of BabelNet synset embedding representations. We also included a system that uses a voting scheme based on several supervised and unsupervised approaches with many different system configurations.

Regarding Task 2 – summarization proper –, we have developed a neural network based on convolutions to learn a specific scoring function. The CNN model was fed by a combination of word embedding with sentence relevance and citation features extracted from each document cluster (RP and CPs).

Acknowledgments

This work is (partly) supported by the Spanish Ministry of Economy and Competitiveness under the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502).

References

1. Abu-Jbara, A., Ezra, J., Radev, D.R.: Purpose and polarity of citation: Towards nlp-based bibliometrics. In: HLT-NAACL. pp. 596–606 (2013)
2. AbuRa'ed, A., Bravo, A., Chiruzzo, L., Saggion, H.: Lastus/taln+ inco@ cl-scisumm 2018-using regression and convolutions for cross-document semantic linking and summarization of scholarly literature. In: Proceedings of the 3nd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL2018). Ann Arbor, Michigan (July 2018) (2018)
3. AbuRa'ed, A., Chiruzzo, L., Saggion, H.: What sentence are you referring to and why? identifying cited sentences in scientific literature. In: RANLP 2017. International Conference Recent Advances in Natural Language Processing; 2017 Sep 2-8; Varna, Bulgaria.[Stroudsburg (PA)]: ACL; 2017. p. 9-17. ACL (Association for Computational Linguistics) (2017)

4. AbuRa'ed, A., Chiruzzo, L., Saggion, H., Accuosto, P., Bravo, À.: Lastus/taln @ clscisumm-17: Cross-document sentence matching and scientific text summarization systems. 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) and co-located with the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017), Tokyo, Japan, August 11, 2017. pp. 55–66 (2017)
5. Camacho-Collados, J., Pilehvar, M.T., Navigli, R.: Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. Artificial Intelligence **240**, 36–64 (2016)
6. Chandrasekaran, M., Radev, D., Freitag, D., Kan, M.Y.: Overview and Results: CL-SciSumm SharedTask 2019. Proceedings of the 4th Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2019) @ SIGIR 2019 (2019)
7. Jaidka, K., Chandrasekaran, M.K., 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)
8. 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 pp. 1–9 (2017)
9. Jaidka, K., Chandrasekaran, M.K., Elizalde, B.F., Jha, R., Jones, C., Kan, M.Y., Khanna, A., Molla-Aliod, D., Radev, D.R., Ronzano, F., Saggion, H.: The computational linguistics summarization pilot task. In: Proceedings of TAC 2014 (2014)
10. Jaidka, K., Yasunaga, M., Chandrasekaran, M.K., Radev, D., Kan, M.Y.: The CL-SciSumm Shared Task 2018: Results and Key Insights. Proceedings of the 3rd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2018) co-located with the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2018) (July 2018)
11. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
12. Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Text summarization branches out: Proceedings of the ACL-04 workshop. vol. 8. Barcelona, Spain (2004)
13. Luhn, H.P.: The automatic creation of literature abstracts. IBM J. Res. Dev. **2**(2), 159–165 (Apr 1958)
14. Ma, S., Zhang, H., Xu, J., Zhang, C.: Njust@ clscisumm-18. In: BIRNDL@ SIGIR. pp. 114–129 (2018)
15. Mihalcea, R., Tarau, P.: Textrank: Bringing order into text. In: Proceedings of the 2004 conference on empirical methods in natural language processing (2004)
16. Navigli, R., Ponzetto, S.P.: BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artif. Intell. **193**, 217–250 (Dec 2012)
17. Nguyen, T.H., Grishman, R.: Relation extraction: Perspective from convolutional neural networks. In: Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing. pp. 39–48 (2015)
18. Nomoto, T.: Resolving citation links with neural networks. Frontiers in Research Metrics and Analytics **3**, 31 (2018)

19. Paice, C.D., Jones, P.A.: The identification of important concepts in highly structured technical papers. In: Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 69–78. SIGIR '93, ACM, New York, NY, USA (1993)
20. Qazvinian, V., Radev, D.R.: Scientific paper summarization using citation summary networks. In: Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1. pp. 689–696. COLING '08, Association for Computational Linguistics, Stroudsburg, PA, USA (2008)
21. Qazvinian, V., Radev, D.R.: Identifying non-explicit citing sentences for citation-based summarization. In: ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden. pp. 555–564 (2010)
22. Ronzano, F., Saggion, H.: Dr. Inventor Framework: Extracting structured information from scientific publications. In: International Conference on Discovery Science. pp. 209–220. Springer (2015)
23. Saggion, H., AbuRa'ed, A., Ronzano, F.: Trainable citation-enhanced summarization of scientific articles. 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. 175–186 (2016)
24. Saggion, H., Lapalme, G.: Concept identification and presentation in the context of technical text summarization. In: Proceedings of the 2000 NAACL-ANLP Workshop on Automatic Summarization. pp. 1–10. Association for Computational Linguistics, Stroudsburg, PA, USA (2000)
25. Saggion, H., Lapalme, G.: Generating indicative-informative summaries with sumum. Comput. Linguist. **28**(4), 497–526 (Dec 2002)
26. Zeiler, M.D.: Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)