University of Mannheim @ CLSciSumm-17: Citation-Based Summarization of Scientific Articles Using Semantic Textual Similarity

Anne Lauscher[1], Goran Glavaš[1], and Kai Eckert[2]

[1] University of Mannheim, Data and Web Science Research Group,
B6 26, 68159 Mannheim, Germany
{anne,goran}@informatik.uni-mannheim.de
[2] Stuttgart Media University, Web-based Information Systems and Services,
Nobelstraße 10, 70569 Stuttgart, Germany
eckert@hdm-stuttgart.de

Abstract. The number of publications is rapidly growing and it is essential to enable fast access and analysis of relevant articles. In this paper, we describe a set of methods based on measuring semantic textual similarity, which we use to semantically analyze and summarize publications through other publications that cite them. We report the performance of our approach in the context of the third CL-SciSumm shared task and show that our system performs favorably to competing systems in terms of produced summaries.

Keywords: Scientific Publication Mining, Scientific Summarization, Information Retrieval, Text Classification

1 Introduction

Citations play an important role in the interpretation of scientific literature and help retrace the evolution of scientific ideas. The text surrounding the citation often reveals some important aspect of the cited publication, e.g., the purpose, polarity, or function (e.g., method or hypothesis) of the citation [1, 10, 8].

The *citances* of a publication, i.e., the sentences of the citing articles containing the citation to the publication in focus [17], are often very useful for higher level analyses of referenced publications, and may contribute to better summarization of scientific articles. The Computational Linguistics Scientific Document Summarization Shared Task (CL-SciSumm) has been designed precisely to encourage the exploitation of citing contexts in automatic summarization of scientific publications [8, 6]. The overall aim of the shared task can be summarized as follows: Given a referenced paper (RP) and a set of its citing papers (CPs), create a (community) summary of the RP. The overall task is divided into the following subtasks:

1a) For each citance, retrieve the RP text span to which the citation refers;

1b) Assign to every citation one or more discourse facets (*Method*, *Aim*, *Result*, *Implication*, and *Hypothesis*), based on the retrieved RP text (result of 1a);
2) Summarize (max. 250 words) the RP, using RP spans retrieved for all citances (with assigned discourse facets), i.e., using the results from 1a) and 1b).

Similar to most systems from previous task editions, we frame (1a) as an information retrieval (IR) task. Given the citance, we rank all RP sentences according to their relevance for the citance. We train a learning to rank (L2R) model with features indicating lexical overlap and semantic similarity between sentences. We then augment the top-ranked RP sentence with its adjacent RP sentences, if they also appear high in the L2R model's ranking. For the discourse facet classification task (subtask 1b), we train one binary classifier for each label. We experimented with Support Vector Machines (SVM) [22] and Convolutional Neural Network (CNN) [11] as learning models. Finally, we generate the summary of the RP by (1) clustering the RP segments retrieved for individual citances according to their semantic textual similarity, and (2) selecting the most informative sentence from each cluster, according to the TextRank score [14]. The official shared task evaluation results show that our system performs favorably to competing systems in terms of quality of the produced summaries.

2 Related Work

Here, we briefly discuss the best performing systems from the previous editions of the CL-SciSumm shared task [7, 8].

Moraes et al. [16] propose two methods for detecting RP spans corresponding to citances: (1) the cosine similarity between the citance and RP candidate text's sparse TF-IDF vectors (2) SVM with tree kernels. Surprisingly, the simple cosine similarity between bag-of-words (BoW) vectors performed better, but the authors still summarized based on the SVM tree kernel ranking.

Li et al. [12] combine lexical overlap and semantic similarity scores (e.g., bag-of-words similarity, unigram overlap, and cosine between word2vec vectors) in a rule-based fashion to select the RP text spans for citances. For summarization, the authors cluster the candidate sentences using hierarchical LDA and compute many features to select cluster representatives for the summary.

On the other hand, Conroy et al.'s [4] summarization method is based on a vector space model, in which they use term frequency and nonnegative matrix factorization to obtain term weights which they then use to create a summary.

3 Methodology

In this Section, we provide methodological details of the approaches we used for solving different subtasks of the shared task.

3.1 Task 1a: Retrieval of Referenced Text Spans

We cast the identification of the referenced RP text span for the CP citance as an IR task, divided into two steps:

1. Ranking RP sentences according to their relevance for the citance;
2. Selecting the sentences for the RP span, based on the above ranking.

Ranking of Candidate Sentences. We resort to the supervised L2R paradigm. Concretely, we train the Coordinate Ascent model optimizing the mean average precision (MAP) from the RankLib library[3], with the following features:

Lexical similarity features. Two features capture the lexical overlap between an RP sentence and a CP citance: (1) *vector space similarity* (VSS) is the cosine between TF-IDF-weighted BoW vectors; (2) *unigram overlap* (UO) is the Jaccard coefficient computed over term sets of the RP sentence and the citance.

Semantic similarity features. An RP sentence can be semantically similar to the citance, but with little or no lexical overlap. We exploit word embeddings (i.e., semantic word vectors) to compute two measures of semantic textual similarity:

Aggregate sentence embedding similarity (AGG) is the cosine between the aggregate sentence embeddings. The aggregate embedding vector of a sentence is obtained simply as the weighted average (with TF-IDF scores of terms as weights) of embeddings of the terms that the sentence contains.

Word mover's similarity (or distance, WMS) [9] is the measure of semantic similarity that aims to compute the maximal similarity (i.e., minimal distance) in meaning between two texts. Let $A \in \mathcal{R}^2$ be the matrix in which rows denote set of unique tokens S of some RP sentence s, and columns represent set of distinct tokens C of the citance c. The WMD score is then the solution to the optimization problem

$$WMD(s,c) = \max_{A} \sum_{i \in 1}^{|S|} \sum_{j \in 1}^{|C|} A_{i,j} \cdot sim(w_i, w'_j),$$

subject to the constraints

$$\sum_{j=1}^{|C|} A_{i,j} = freq(w_i, s), \;\; \forall i \in \{1, \ldots |S|\}, \text{and}$$

$$\sum_{i=1}^{|S|} A_{i,j} = freq(w'_j, c), \;\; \forall j \in \{1, \ldots |C|\},$$

[3] Online available at: `https://sourceforge.net/p/lemur/wiki/RankLib/`.

where $sim(w_i, w'_j)$ is the cosine similarity of embedding vectors of words w_i and w'_j and $freq(w, s)$ is the frequency with which the word w appears in sentence s. We compute both of the above features (AGG and WMS) using two different sets of word embedding vectors. We experiment with (1) 300-dimensional Skip-Gram embeddings [15], pre-trained on the Google News dataset[4] and (2) 300-dimensional domain-specific embeddings obtained by running the CBOW model [15] on the ACL Reference Corpus [2].

Entity-based features. We run the TagMe entity linker [5] over the citances and the RP candidate sentences to link mentions to Wikipedia concepts. We compute the *entity overlap* (EO) feature as the Jaccard coefficient over sets of linked entities from the citance and the RP sentence. We add a binary feature indicating whether the RP sentence contains any linked entities.

Positional features. We compute the relative sentence position (absolute position normalized by the document length) for the candidate RP sentence in the RP and for the citance in the CP. Similarly, for both the RP candidate within the RP and citance within the CP we extract the relative section positions (section number of the sentence divided by the total number of sections). Finally, we compute the ratio between relative positions of RP sentence and CP citance.

Adjacency-Based Postprocessing. Our L2R model ranks individual RP sentences. Although most often the relevant RP texts of citances have one sentence, reasonably often they also contain two or more sentences. To account for such cases, we perform a postprocessing step where we decide whether to add additional sentences to the output. We evaluated three postprocessing strategies:

1. *Top-rank* returns the top-ranked sentence from the L2R model's ranking;
2. *Top-K neighbours* extends the output with RP sentences adjacent to the top-ranked sentence if these are found within the K top-ranked sentences in the L2R model's ranking;
3. *Iterative Top-K neighbours* extends the *Top-K neighbours* by repeatedly adding adjacent sentences of those already in the output if adjacent sentences are among the top K in the ranking.

3.2 Task 1b: Discourse Facet Classification

The second subtask (subtask 1b), discourse facet classification, is a multi-label classification task. Each RP text span retrieved as relevant for a citance needs to be annotated with appropriate discourse facet labels. Since the retrieved RP text snippet may be labeled with more than one discourse facet, we train one binary classifier for each discourse facet label. For each of the five binary classification tasks, we experimented with two supervised machine learning models: Convolutional Neural Networks (CNN) and Support Vector Machines (SVM).

[4] Available at `https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit?usp=sharing`.

CNNs [11], first applied on NLP tasks by Collobert and Weston [3], have been shown to be successful on a range of short text classification task (see [21, 20], *inter alia*). The architecture of the CNN we apply consists of a single convolutional layer, followed by a single max-pooling layer. We use rectified linear unit (ReLU) as the non-linear activation function. In addition to the vanilla CNN model that makes predictions based purely on the retrieved RP text span, we also evaluate a CNN variant in which we introduce hand-crafted features that are, for each instance, concatenated to the latent CNN features (i.e., the output of the max-pooling layer) and fed to the last (feed-forward) layer of the network, which makes the final label prediction. Let $\mathbf{x}_{CNN}$ be the latent CNN vector for some input example, and $\mathbf{x}_{HF}$ be the vector of hand-crafted features for the same RP text span instance. The output vector $\mathbf{y}$ (a probability distribution over the two labels in binary classification tasks) is then computed as follows:

$$\mathbf{y} = softmax\left(\mathbf{W} \cdot (\mathbf{x}_{CNN} \| \mathbf{x}_{HF}) + \mathbf{b}\right),$$

where $\mathbf{W}$ and $\mathbf{b}$ are the weights matrix and biases vector of a feed-forward network with a single hidden layer and linear activation. For the CNN classification tasks we represent the input tokens with 300-dimensional domain-specific word embeddings, trained on the ACL Reference Corpus [2] (see Section 3.1).

We also experimented with binary SVM [22] classifiers, employing the following set of hand-crafted features (all features except lexical are also used as additional hand-crafted features for the hybrid CNN model):

Lexical features. The sparse TF-IDF weighted BoW vector of the RP span;

Positional features. The relative sentence position and the relative section position of the retrieved RP text span;

Other features. Two binary features indicating whether the retrieved RP span (1) contains numbers and (2) consists of multiple sentences.

3.3 Task 2: Citation-Based Summarization

Finally, we create the RP summary by exploiting the output of the first task – the retrieved RP text spans for all citances. To build a non-redundant summary reflecting the most important aspects of the RP, we propose the following rule-based approach:

1. We cluster the RP text spans (retrieved for all of the citances) using the simple single-pass clustering algorithm employing word mover's similarity (cf. Section 3.1) as the similarity score between different RP text spans;
2. In order to select the most informative sentences for the summary, we compute the TextRank score [5] for each retrieved RP span and order the RP spans within clusters according to their TextRank scores;
3. We rank the clusters according to the average TextRank scores of the RP text spans they contain. We then first select for the summary the most informative text span from the most informative cluster, then the most informative text span from the second most informative cluster, etc., until we reach the summary limit of 250 words.

4 Evaluation

We first describe the datasets we used to train and optimize our models. Next, we explain the evaluation setting and describe different configurations we submitted for the final evaluation.

4.1 Dataset

The training set provided by the shared task organizers[5] is an annotated subset of the ACL Anthology [19, 18] consisting of 30 topics, each of which consists of one referenced paper (RP) and its corresponding citing papers (CPs). In total, the training set consists of 594 instances, i.e., citances paired (i.e., annotated) with relevant references text spans (indicated as sentence offsets in the RP) and the corresponding discourse facet labels.

4.2 Evaluation Setting

For subtask (1a), i.e., retrieval of relevant RP spans for given citances, we evaluated different model variants via the 10-folded cross validation (CV) on the training set. Positive instances for the L2R model are given directly as citance–RP text span pairs. For the negative instances, one could couple the citance with any other portion of text from the RP. In order to prevent excessive skewness of the training set in favor of the negative examples, we sampled 10 negative instances for each positive instance, picking both sentences adjacent to the relevant RP text span and RP sentences from other article sections. We optimized the L2R model for mean average precision (MAP), i.e., we searched (via greedy feature selection) for the combination of features with the largest MAP performance.

To evaluate the effects of different postprocessing strategies (see Section 3.1), we ran the evaluation script provided by the task organizers on outputs produced by different RP span selection models (we used the gold discourse facet labels in this case). We estimated our performance on the discourse facet classification task in a 5-fold CV setting in terms of precision, recall and F1-score, micro-averaged over the folds. Finally, our system's output for the final summarization task was evaluated, also in CV setting on the train set, in terms of the ROUGE-2 score against three types of gold summaries – RP abstract, expert human summary, and community summary [13]. We experimented with different WMS thresholds for the single-pass clustering. We omit the CV performance on the train set due to space constraints.

4.3 Submitted Runs

In total, we submitted 9 different runs, composed as follows:

[5] Online available at `https://github.com/WING-NUS/scisumm-corpus/tree/master/data/Training-Set-2017`.

Task (1a). According to CV evaluation on the training set, the best feature combination consisted of: unigram overlap (UO), vector space similarity (VSS), aggregated embedding similarity using domain-specific word embeddings (AGG-ACL), and WMS based on the domain-specific word embeddings (WMS-ACL). We coupled the predictions of the L2R model trained with this feature combination with three post-processing strategies: *top-rank*, *top-5 neighbours*, and *top-10 neighbours*. This gave us three runs for retrieving relevant RP spans.

Task (1b). For the discourse facet classification we considered three variants: (1) predict with SVM classifiers for all five facet labels, (2) predict with CNN classifiers for all five labels, and (3) for each label, predict with the classifier that yielded best results in the CV setting on the training set. These three variants, combined with three retrieval variants for (1a) resulted in total of nine runs for tasks 1a and 1b together.

Task (2). CV experiments on the training set suggested the value of 0.85 to be the optimal WMS threshold for the single-pass clustering. Using only the results of subtask (1a) for summarization (i.e., our summarization algorithm does not use discourse facets), we submit three summaries for each topic.

4.4 Final Results

The final evaluation of the submissions was performed by the organizers of the shared task. Here, we report the results of our submitted runs as well as the average and winning scores across all participants. For more information please refer to the overview paper of the shared task [6].

The results of the referenced text span identification task (task 1a) are listed in table 1. Just picking the top-ranked sentence outputted by our L2R model is numerically above the average performance across all submissions. Our best result is reached using the *top-5 neighbors* postprocessing, i.e., by searching among the top-5 ranked candidate sentences for neighbors of the top-ranked sentence. Due to the very limited size of the test set, the performance differences are most likely not statistically significant.[6]

Table 2 shows the results of task (1b), the discourse facet classification. The results heavily depend on the output of task 1a, thus it is not surprising that the classifications produced on top of the sentences retrieved using our L2R model with the *top-5 neighbors* postprocessing strategy exhibit best for all three classification strategies applied. The best score was achieved by applying only SVM classifiers. The limited size of the provided training data is the most likely explanation for the CNN classifiers performing worse than the SVM classifiers.

The evaluation of the final output summaries performed by the organizers (see table 3) shows that our approach performs best compared to those of the other participants in terms of ROUGE-SU4 F1 score when compared against the *community* and the *abstract* gold summaries. Moreover, for all three variants of

[6] The shared task organizers provided no information on statistical significance of the performance differences between submissions.

Table 1. Results of the referenced text span identification task on the test set (task 1a, ROUGE-2, in %).

Strategy	Precision	Recall	F1
Top-ranked	06.3	11.5	07.2
Top-5 neighbors	07.6	11.4	07.5
Top-10 neighbors	08.9	09.7	06.8
Average Score	20.2	07.0	07.1
Winning Score	37.0	13.2	11.4

Table 2. Results of the discourse facet classification task on the test set (task 1b, macro average, in %).

Strategy	Precision	Recall	F1
Top-ranked + CNN	20.0	06.3	09.6
Top-ranked + Hybrid	20.8	06.3	09.7
Top-ranked + SVM	25.8	06.3	10.2
Top-5 neighbors + CNN	25.0	07.2	11.1
Top-5 neighbors + Hybrid	25.8	07.2	11.2
Top-5 neighbors + SVM	30.8	07.2	11.6
Top-10 neighbors + CNN	23.8	07.5	11.4
Top-10 neighbors + Hybrid	23.3	06.3	10.0
Top-10 neighbors + SVM	28.3	06.3	10.3
Winning Score	93.8	28.9	40.8
Average Score	47.0	13.7	20.8

the gold summaries – human expert summary, author abstract, and community summary – our approaches reach the highest precision. These results suggest that the errors in identifying the referenced text spans (task 1a) do not necessarily propagate to summary composition, which, in turn, suggests that referenced RP texts might not be the best source of text for constructing the summaries.

5 Conclusion

In this work, we presented a combination of methods for citation-based semantic analysis and summarization of scientific publications. For the retrieval of referenced text spans we employed a supervised learning to rank model with a number of features capturing semantic textual similarity between the citation context and reference paper sentences. Next, we experimented with SVM and CNN classifiers for discourse facet classification. Finally, we proposed a simple summarization approach based on clustering of the referenced sentences, again by exploiting measures of semantic textual similarity. The official evaluation of the automatically created publication summaries shows that our system produces higher quality than competing systems in several evaluation settings.

Table 3. Results of the summarization task on the test set (task 2, ROUGE-SU4, in %). Scores are in bold if the number corresponds to the winning score.

Strategy	vs. Abstract			vs. Human			vs. Community		
	P	R	F1	P	R	F1	P	R	F1
Top-ranked	**13.9**	35.3	**19.1**	38.8	11.7	16.6	**18.9**	20.3	**17.4**
Top-5 neighbors	13.6	34.5	18.7	**39.3**	11.9	16.9	18.4	18.6	16.7
Top-10 neighbors	12.8	36.1	18.4	35.2	11.2	15.7	17.9	20.1	16.9
Average Score	10.1	34.7	15.0	28.2	10.6	14.1	14.5	17.8	14.5
Winning Score	13.9	51.2	19.1	39.3	14.4	17.8	18.9	23.8	17.4

Acknowledgements

This research was partly funded by the German Research Foundation (DFG), grant number EC 477/5-1 (LOC-DB). We thank the NVIDIA Corporation for donating the GeForce Titan X GPU used to carry out some of our experiments.

References

1. Abu-Jbara, A., Ezra, J., Radev, D.: Purpose and polarity of citation: Towards nlp-based bibliometrics. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HTL). pp. 596–606. ACL (2013)
2. Bird, S., Dale, R., Dorr, B., Gibson, B., Joseph, M., Kan, M.Y., Lee, D., Powley, B., Radev, D., Tan, Y.F.: The acl anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC). pp. 1755–1759 (2008)
3. Collobert, R., Weston, J.: A unified architecture for natural language processing: deep neural networks with multitask learning. In: Cohen, W.W., McCallum, A., Roweis, S.T. (eds.) Proceedings of the 25th International Conference on Machine Learning (ICML). pp. 160–167 (2008)
4. Conroy, J., Davis, S.: Vector space and language models for scientific document summarization. In: Proceedings of NAACL-HLT. pp. 186–191 (2015)
5. Ferragina, P., Scaiella, U.: Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM). pp. 1625–1628. ACM, New York, NY, USA (2010)
6. Jaidka, K., Chandrasekaran, M.K., Jain, D., Kan, M.Y.: Overview of the cl-scisumm 2017 shared task. In: Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries 2017 (BIRNDL). CEUR, Tokyo, Japan (2017)
7. Jaidka, K., Chandrasekaran, M.K., Elizalde, B.F., Jha, R., Jones, C., Kan, M.Y., Khanna, A., Radev, D.R., Ronzano, F., Saggion, H., Kim, W.: The computational linguistics summarization pilot task (2014)

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 (Jun 2017)
9. Kusner, M., Sun, Y., Kolkin, N., Weinberger, K.: From word embeddings to document distances. In: Bach, F., Blei, D. (eds.) Proceedings of ICML 2015. vol. 37, pp. 957–966. PMLR, Lille, France (07–09 Jul 2015)
10. Lauscher, A., Glavaš, G., Ponzetto, S.P., Eckert, K.: Investigating convolutional networks and domain-specific embeddings for semantic classification of citations. In: Proceedings of the Workshop on Mining Scientific Publications (WOSP '17). p. in press. ACM (2017)
11. LeCun, Y., Bengio, Y.: The handbook of brain theory and neural networks. chap. Convolutional Networks for Images, Speech, and Time Series, pp. 255–258. MIT Press, Cambridge, MA, USA (1998)
12. Li, L., Mao, L., Zhang, Y., Chi, J., Huang, T., Cong, X., Peng, H.: Cist system for cl-scisumm 2016 shared task. In: Cabanac, G., Chandrasekaran, M.K., Frommholz, I., Jaidka, K., Kan, M.Y., Mayr, P., Wolfram, D. (eds.) Proceedings of BIRNDL 2016. CEUR Workshop Proceedings, vol. 1610, pp. 156–167. CEUR-WS.org (2016)
13. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Marie-Francine Moens, S.S. (ed.) Text Summarization Branches Out: Proceedings of the ACL-04 Workshop. pp. 74–81. ACL, Barcelona, Spain (July 2004)
14. Mihalcea, R., Tarau, P.: Textrank: Bringing order into texts. In: Lin, D., Wu, D. (eds.) Proceedings of the Conference on Empirical Methods in Natural Language Processing 2004 (EMNLP). pp. 404–411. ACL, Barcelona, Spain (July 2004)
15. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS, pp. 3111–3119. Curran Associates, Inc. (2013)
16. 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: Cabanac, G., Chandrasekaran, M.K., Frommholz, I., Jaidka, K., Kan, M.Y., Mayr, P., Wolfram, D. (eds.) Proceedings of BIRNDL 2016. CEUR Workshop Proceedings, vol. 1610, pp. 113–121. CEUR-WS.org (2016)
17. Nakov, P.I., Schwartz, A.S., Hearst, M.A.: Citances: Citation sentences for semantic analysis of bioscience text. In: In Proceedings of the SIGIR04 workshop on Search and Discovery in Bioinformatics (2004)
18. Radev, D.R., Muthukrishnan, P., Qazvinian, V.: The ACL anthology network corpus. In: Proceedings, ACL Workshop on Natural Language Processing and Information Retrieval for Digital Libraries. pp. 54–61. Singapore (2009)
19. Radev, D.R., Muthukrishnan, P., Qazvinian, V., Abu-Jbara, A.: The acl anthology network corpus. Language, Resources and Evaluation 47(4), 919–944 (Dec 2013)
20. Severyn, A., Moschitti, A.: Twitter sentiment analysis with deep convolutional neural networks. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 959–962. SIGIR '15, ACM, New York, NY, USA (2015)
21. Shrestha, P., Sierra, S., González, F.A., Rosso, P., Montes-y Gómez, M., Solorio, T.: Convolutional neural networks for authorship attribution of short texts. In: Proceedings of the 2017 Conference of the European Chapter of the Association of Computational Linguistics (EACL) (2017)
22. Vapnik, V.: Estimation of Dependences Based on Empirical Data: Springer Series in Statistics (Springer Series in Statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA (1982)