Poli2Sum@CL-SciSumm-19: Identify, Classify, and Summarize Cited Text Spans by means of Ensembles of Supervised Models

Moreno La Quatra, Luca Cagliero, and Elena Baralis

Politecnico di Torino Corso Duca degli Abruzzi, 24 10129 Turin (Italy)
{name.surname}@polito.it

Abstract. This paper presents the Poli2Sum approach to the 5th Computational Linguistics Scientific Document Summarization Shared Task (BIRNDL CL-SciSumm 2019). Given a set of reference papers and the set of papers citing them, the proposed approach has a threefold aim. (1a) Identify the text spans in the reference paper that are referenced by a specific citation in the citing papers. (1b) Assign a facet to each citation describing the semantics behind the citation. (2) Generate a summary of the reference paper consisting of the most relevant cited text spans. The Poli2Sum approach to tasks (1a) and (1b) relies on an ensemble of classification and regression models trained on the annotated pairs of cited and citing sentences. Facet assignment is based on the relative positions of the cited sentences locally to the corresponding section and globally in the entire paper. Task (2) is addressed by predicting the overlap (in terms of units of text) between the selected text spans and the summary generated by the domain experts. The output summary consists of the subset of sentences maximizing the predicted overlap score.

Keywords: Citation identification · Sentence-based summarization · Classification and Regression · Text mining

1 Introduction

The diffusion of digital libraries has eased the access to scientific publications in electronic form. The paper full-text, the author and co-author relationships, and the citation networks have become accessible from the most popular Web-based sources. For example, DBLP [12] is a computer science bibliography providing online reference for open bibliographic information on computer science journals. In parallel, networks for linking scientists and researchers (e.g., ResearchGate, Academia.edu [21]) as well as online services to index, search, and mine scientific data at large (e.g., ArnetMiner [23]) have been developed.

Since exploring the textual content of scientific papers is extremely time-consuming, the recent advances in Information Retrieval and Computational Linguistics have focused on automating the process of knowledge extraction and linking from scientific papers and related social data. The main challenges

addressed in the research community include, amongst others, modeling author-topic relationships (e.g., [19]), identifying of cross-topic collaborations (e.g., [14]), and detecting potential conflicts of interest (e.g., [24]).

The Computational Linguistics Scientific Document Summarization Shared Task (CL-SciSumm 2019) [4] presented at the joint workshop on Bibliometric-enhanced IR and NLP for Digital Libraries (BIRNDL@SIGIR 2019) [9] is a challenge focused on text mining and summmarization of scientific papers. It considers topics described by a reference papers and a set of citing papers that all contain citations to the reference paper. In each citing paper, the text spans (i.e., citances) have been identified that pertain to a particular citation to the RP. The task proposed in the 5th edition of the challenge, i.e., CL-SciSumm BIRNDL 2019, entails automatically generating summaries of scientific papers of two types: faceted summaries of the traditional self-summary (the abstract) and the community summary (the collection of citation sentences). Furthermore, it entails grouping the citation sentences by the facets of the text that they refer to. A more formal statement of the CL-SciSumm 2019 Shared Task[1] is given below.

Problem statement. Let rp be a reference scientific paper and let CP be the set of scientific papers citing in rp (hereafter denoted as *citing papers*). Given an arbitrary citing paper $cp \in CP$ let $c_{CP}=\{c_1, c_2, \ldots, c_n\}$ be the text spans in cp (hereafter denoted as *citances*) pertinent to any citation to rp (i.e., the parts of the text where citations to rp are placed).

The tasks can be formulated as follows.

(1A) For each citance c_j, identify the spans of text $rp(c_j)$ in the reference paper (hereafter denoted as *cited text spans*) that are most likely to be related to c_j. The cited text spans can be either a single sentence or consecutive sentences (no more than 5).

(1B) For each cited text span identify what facet of the paper it belongs to from a predefined set of facets. Facets describe the semantics behind the citation (i.e., Aim, Hypothesis, Implication, Results, Method).

(2) Produce a short summary of the reference paper (no more than 250 words) which consists of a selection of cited text spans (this task is optional).

This paper presents the Poli2Sum approach[2] to the 5th Computational Linguistics Scientific Document Summarization Shared Task (BIRNDL CL-SciSumm 2019). Our approach relies on an ensemble of classification and regression models trained on the annotated pairs of cited and citing sentences. Supervised models are trained on a variety of features, including those extracted by using two among the most popular word embedding models (i.e., Word2Vec [15], Sent2Vec [16]). Two complementary predictive variables have been considered in the proposed

[1] http://wing.comp.nus.edu.sg/ cl-scisumm2019/

[2] The name of the method is an acronym abbreviating the name of the university to which the authors are affiliated (Politecnico di Torino, Italy) and the keyword *Summarization*.

approach: (i) a discrete label, indicating whether a particular citance refers to given cited text span, and (ii) a continuous class label, which indicates, for a given citance, the distance between the candidate text span and the actual cited text span. To forecast the value of the predictive variables, ensembles of classification and regression methods have been applied, respectively. Classification models are aimed at accurately predicting the discrete class label associated with each pair of citance and cited text span. Regression models are instead applied to produce a rank of the cited text spans associated with a given citance. The most likely text span is the one that minimizes the distance [1].

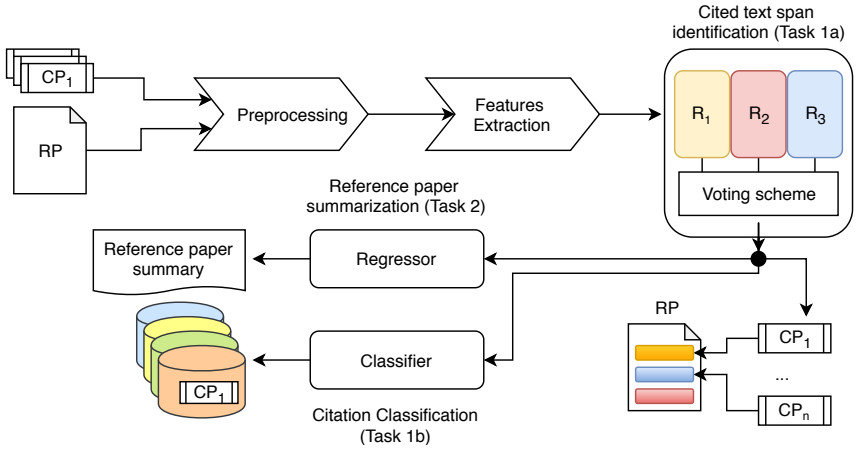
The Poli2Sum approach to tackling tasks 1A and 1B relies on an ensemble of classification and regression models trained on the annotated pairs of cited and citing sentences. To generate an ensemble of binary classifiers for task 1A, the outcomes of the single classifiers are properly combined using a greedy strategy in order to select the text spans with maximal number of votes. Thanks to the inherent rank given by the regressors, the ensemble of regression models for Task 1A identifies the text spans that minimize the predicted distance. To tackle facet assignment (i.e., task 1B), a classification model is trained on the annotated subset of text spans. The annotated data are enriched with additional information about the structure paper (i.e., the relative positions of the cited text span in the section and in the entire paper). The idea behind is that the semantics of the citation is likely to be correlated with the position of the cited text span. Finally, to tackle task 2 a regression model predicts the overlap, in terms of units of text, between the selected text spans and the manually generated summaries. The more similar the content of the text span with the human annotation, the more likely the content is worth including in the summary. The output summary is a selection of the top ranked text spans in order of decreasing overlap score.

The rest of the paper is organized as follows. Section 2 compares the Poli2Sum approach with the previously proposed solutions. Section 3 thoroughly describes the Poli2Sum method, while Section 4 summarizes the main experimental results. Finally, Section 5 draws conclusions and discusses future works.

2 Related works

In the previous editions of CL-SciSumm 2019 Shared Task many effective strategies to tackle the aforesaid problems have been proposed [8]. For example, the best performing approach to Task 1A in the former edition [1] proposed to use Convolutional Neural Networks and word Embeddings. They designed a voting system combining supervised (CNNs) with unsupervised (WEs) models. Similar to [1], we propose to use regression models to predict the distance between the actual and candidate cited text spans. Unlike [1], the Poli2Sum approach relies on ensemble methods combining either classification or regression models. Furthermore, word embedding information is integrated as training data features instead of used to drive separate strategies. The systems proposed in [6, 20] rely on binary classification models which consider more advanced textual features extracted using word embedding and Natural Language Processing techniques.

Fig. 1: Poli2Sum architecture.



They considered four main feature categories: similarity features, positional features, frequency-based features, and rule-based features. Unlike [6, 20] this work exploits regression models as well. Furthermore, in [6] the authors did not address the optional summarization task (i.e., task 2). Other approaches (e.g., [17]) exploited various contextualized word vector spaces trained either on GoogleNews[3] or on the ACL Antology Network[4]. More advanced text distance measures computed at the levels of words or sentences have also been proposed, i.e., the Word Movers Distance and the Earth Movers Distance [10, 7], the IDF-weighted Average Embedding based similarity and the Smooth Inverse Frequency based similarity distances [7]). Unlike Poli2Sum, in [7] task 2 has not been addressed, while in [10] the proposed solution is not based on supervised machine learning techniques.

3 The Poli2Sum method

The architecture of the Poli2Sum system is depicted in Figure 1. A brief summary of the functionality of each block is given below, while more extensive descriptions are provided in the following sections.

- **Parsing and preprocessing:** this block is devoted to parsing and preparing the raw text of the analyzed documents to the subsequent steps.
- **Cited text span identification:** It trains and applies ensembles of regression or classification models to specifically address *task 1a*.
- **Citation classification:** it addresses *task 1b* by training and applying a multi-class classification model on top of the outcomes of the *Cited text span identification* step.

[3] https://google.news.com
[4] http://clair.eecs.umich.edu/aan/index.php

– **Reference paper summarization:** this step entails ranking the cited text spans selected by the *cited text span identification* step to generate the reference paper summary (output of task 2). It applies a supervised model to predict the significance level of each text span according to the training data.

The text preparation and analytics steps performed by Poli2Sum are implemented in the Python language and rely on machine learning models provided by the Scikit-learn library [18].

3.1 Parsing and preprocessing

The text of the scientific papers and the related citation network are processed in order to tailor the input data to the subsequent analyses. First, a parsing of the text, provided in xml format, is performed by considering also the text structure (e.g., the organization of the text into sentences and sections). Then, the input text is tokenized into separate words and the less relevant or non-informative words (e.g., conjunctions, prepositions) are removed. Word tokenization and stop-word removal were based on English vocabulary provided by the Natural Language Toolkit [2]. Finally, the text is also transformed into word and sentence latent spaces using the Word2Vec [15] and Sent2Vec [16] algorithms, respectively. The word embeddings are trained using the Wikipedia corpus recommended by the authors [15].

3.2 Cited text span identification (Task 1a)

For each citance, the goal of this block is to identify the span of referenced text. Among the candidate sentences in the reference paper, the aim is to exploit the content of the citing snippet and the semantics behind the candidate sentence. Furthermore, since the cited text span may include more than one sentence (at most 5), a parallel issue is to decide whether a sequence of sentences is worth considering instead of separate individual sentences.

To tackle the above issues, it creates a structured training dataset containing one record for each pair of candidate cited text span and citance. The dataset features consists of a variety of measures evaluating the similarity between citance and candidate text span. The considered features are enumerated below.

– **Sent2Vec similarity:** it indicates the distance between the citance and the candidate text in the latent space of the document sentences. It is computed as the cosine similarity between the corresponding latent vectors generated by the Sent2Vec embedding model proposed by [16].
– **Word2Vec similarity:** it indicates the distance between the citance and the candidate text in the latent space of the document words. It is computed as the cosine similarity between the corresponding vectors generated by the Word2Vec embedding model [15]. To generate sentence vectors, single word vectors are averaged and stop-words are excluded.

– **Rouge-based similarities:** they indicate the syntactical similarity between the citance and the candidate text span. It is computed as the F-measure score produced by the established Rouge toolkit [13]. To analyze overlaps among text at different granularity levels, the considered units of overlap are: (i) unigrams (i.e., Rouge-1), (ii) bigrams (i.e., Rouge-2), and (iii) the largest matching sub-sequence (Rouge-L).

The content of the training dataset is extracted from the CL-SciSumm 2019 training data[5]. Two complementary solutions, respectively based on classification and regression models, have been integrated.

Classification-based approach Each record of the structured dataset is labeled as (i) *true*, if the sentence of the reference paper is actually part of the cited text span, or (ii) *false* otherwise. To cope with imbalances in the training data, the structured dataset contains all the records labelled as *true* and 7 sample records labeled with *false* (those with maximal Sent2Vec similarity score). The idea behind to train robust prediction models for the *true* class label by considering the most challenging instance for *false* class label.

An ensemble of three different classification models with different characteristics is trained. Specifically, it considers a decision tree-based model (i.e., Gradient Boosting), a Neural Network (i.e., Multi-Layer Perceptron), and a Bayesian classifier (i.e., Gaussian Naive Bayes) [11]. Considering heterogeneous model type increases the chance to capture different correlations between the dataset feature and the class. Separately for each model, a bagging model averaging approach [3] is applied in order to make the system more robust to small variations between the training and validation sets.

Regression-based approach The records in the structured dataset are labeled by assigning the maximum Sent2Vec similarity score between the candidate sentence of the reference paper and the set of sentences actually referenced by the citances. Notice that (i) the similarity is maximal if the candidate text span is actually referenced by the citance (1), and (ii) the target value expresses the similarity between the candidate text span and the closest text span actually referenced by the citance in the embedding space. Similar to the classification approach, an ensemble of three regression algorithms (i.e., Gradient Boosting, Multi-Layer Perceptron and Adaboost [11]) is trained and a bagging model averaging approach is applied. The selection of the candidate text span is given by a majority voting procedure. For each regressor, the 7 top scored sentences are picked first. Then, a majority voting process is used to pick the most relevant sentences by consensus among all the considered regressors.

3.3 Citation classification (Task 1b)

A multi-class classification model, based on the Gradient Boosting algorithm, is trained in order to assign a facet to each citation. A citation consists of a

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

pair of citance and cited text span. The class labels are the facets describing the semantics behind the citation (i.e., Aim, Hypothesis, Implication, Results, Method). The training dataset with annotated citations is given by the contest organizers. The *Citation classification* block enriches the training dataset with positional features in order to take the relative position of the cited text span into account during the facet assignment process. The following features are considered.

- **Position of the referenced text:** we consider the top-ranked referenced sentence and we compute a normalized score, between 0 and 1, using the sentence identified. Specifically, we compute the ratio between the predicted sentence position (s_p) and the length, in terms of sentences, of the paper (s_{max}).
- **Section-based features:** when the title of the section is available, we use a regular expression based method to divide the paper sections in 7 classes, namely Title, Abstract, Introduction, Related Works, Method Description, Experiments, Conclusion. For each sentence, the information about the parent section is encoded as a discrete feature. Moreover, when available, we used the section numbering directly as additional feature of the system.

3.4 Summarization (Task 2)

The summarization task entails generating a concise yet informative summary of the reference paper consisting of the most relevant cited text spans. Hence, this module aims at evaluating the cited text span in order to identify the best representatives. To tackle this issue, a ground truth has been provided by the contest organizers. It contains a subset of reference papers annotated with a manually generated summary. This block trains a regression model on the annotated sentences by considering for each cited text span the following set of describing features:

- **Sentence length:** it indicates the number of words in the sentence. The longer the sentences the more likely the sentence would contain repetitions or redundant information.
- **Embedding-based similarities with the abstract:** they indicate the similarity between the candidate text span and the abstract of the paper, computed using both *Word2Vec* [15] and *Sent2Vec* [22] embedding methods. Since the abstract could be deemed as a short summary of the paper, it could be helpful for discriminating between relevant text spans and not.
- **Syntactical similarities with title and abstract:** they indicate the syntactical similarity between the candidate text span and the content of the title and the abstract of the paper, respectively. It is computed using the F-measure of the Rouge-L metric (i.e., we maximize the overlap in terms of longest matching sub-sequence). The same procedure is applied between each sentence and the paper's abstract. This metric is able to identify the longest common sub-sequence of words between two text spans.

The target of the regression model, trained using the Gradient Boosting algorithm, is the Syntactical similarity between the candidate text span and the humanly generated summary (in terms of Rouge-L F-measure).

4 Experiments

To evaluate the performance of the Poli2Sum approach, we followed the recommendations provided by the CL-SciSumm-19 task organizers and we used the following datasets for Tasks 1.a and 2:

- **Training set:** We used the training data provided by the CL-SciSumm-19 task organizers.
- **Validation set:** We used the training dataset of CL-SciSumm-18 (40 papers).
- **Test set:** We applied the trained models on the test data provided by the CL-SciSumm-19 task organizers.
- **Ground truth:** The test outcomes are compared with the ground truth by the CL-SciSumm-19 task organizers.

Since the CL-SciSumm-19 training data are unlabeled, to tackle Task 1.B we applied an hold-out validation strategy (75% of the dataset was used for training, while the remaining part for validation).

The hyper-parameters of the regression and classification algorithms were tuned on the training set using a 5-fold cross-validation procedure. The parameter settings that differ from the standard recommendations provided by the SciKit-learn library [18] are reported below.

- Task 1.A: Gradient Boosting regressor (num. estimators = 200), AdaBoost Regressor (num. estimators = 200), MultiLayer Perceptron (num. of layers = 2, layer size = 100)
- Task 1.B: Gradient Boosting classifier (num. estimators = 100)
- Task 2: Gradient Boosting regressor (num. estimators = 400)

4.1 Results of Task 1.A (Cited text span identification) on the validation set

We performed an ablation study to assess the performance improvements achieved by the single regression methods, the ensemble method, and the bagging strategy. Table 1 reports the results obtained by single regressors. We empirically analyzed the performance of the regression models by varying the number of selected sentences. Setting the number of selected sentence to 5 allowed us to achieve the best results in terms of F-measure.

Table 2 reports the results of the ensemble methods by enabling and disabling the bagging option. Enabling the bagging strategy yielded slight performance improvements. Similar results (not reported here due to the lack of space) were achieved using the classification approach. Regression-based ensemble methods turned out to be slightly more effective than classification-based ones on the tested data (best F1-measure 0.14 vs 0.13).

Num. of sentences	Method	Precision	Recall	F-measure
5	Multi-Layer Perceptron	0.08	0.25	0.12
	AdaBoost	0.08	0.26	0.12
	Gradient Boosting	0.08	0.26	0.12
7	Multi-Layer Perceptron	0.07	0.32	0.12
	AdaBoost	0.07	0.30	0.11
	Gradient Boosting	0.08	0.34	0.13
10	Multi-Layer Perceptron	0.06	0.36	0.10
	AdaBoost	0.06	0.36	0.10
	Gradient Boosting	0.06	0.38	0.11

Table 1: Task 1a: Performance of single regressors on the validation set.

Bagging	Num. of sentences	Precision	Recall	F-measure
Enabled	5	0.10	0.22	0.14
	7	0.09	0.27	0.14
	10	0.08	0.34	0.13
Disabled	5	0.09	0.24	0.13
	7	0.08	0.29	0.12
	10	0.07	0.31	0.12

Table 2: Impact of bagging on regression performance on the validation set.

4.2 Result of Task 1B (Citation classification) on the validation set

Table 3 reports the results obtained for the task 1.B in terms of precision, recall, and F-measure on the validation set.

Type	*Precision*	*Recall*	*F-measure*
Micro average	0.48	0.48	0.48
Macro average	0.16	0.19	0.18
Weighted average	0.40	0.48	0.44

Table 3: Task 1B: Performance of citation classification on the validation set

4.3 Results of Task 2 (Reference paper summarization) on the validation set

We empirically analyzed the performance of the summarization process by comparing the automatically generated summaries with those generated by the domain experts. To perform a quantitative evaluation, we used the standard Rouge toolkit [13]. Table 4 reports the average Rouge-2 and Rouge-L scores (recall,

Type	*Metric*	*Precision*	*Recall*	*F-measure*
Human Summary	Rouge-2	0.198	0.135	0.153
	Rouge-L	0.376	0.283	0.284
Community Summary	Rouge-2	0.204	0.337	0.241
	Rouge-L	0.315	0.505	0.331

Table 4: Task 2: Summary evaluation on the validation set

precision, and F1-measure) achieved against for the *community* and the *human-annotated* summaries. To analyze the qualitative significance of the achieved results, Table 5 reports an example of generated summary as well as the corresponding humanly generated version. The summary generated by Poli2Sum appeared to be fairly consistent with the expected result.

Poli2Sum Summary
This paper presents a corpus-based approach to word sense disambiguation where a decision tree assigns a sense to an ambiguous word based on the bigrams that occur nearby. This approach is evaluated using the sense-tagged corpora from the 1998 SENSEVAL word sense disambiguation exercise. Word sense disambiguation is the process of selecting the most appropriate meaning for a word, based on the context in which it occurs. There is a further assumption that each feature is conditionally independent of all other features, given the sense of the ambiguous word. In particular, the decision tree learner makes decisions as to what bigram to include as nodes in the tree using the gain ratio, a measure based on the overall Mutual Information between the bigram and a particular word sense. We have presented an ensemble approach to word sense disambiguation (Pedersen, 2000) where multiple Naive Bayesian classifiers, each based on co-occurrence features from varying sized windows of context, is shown to perform well on the widely studied nouns interest and line. Bigrams have been used as features for word sense disambiguation, particularly in the form of collocations where the ambiguous word is one component of the bigram (e.g., (Bruce and Wiebe, 1994), (Ng and Lee, 1996), (Yarowsky, 1995)). The results of this approach are compared with those from the 1998 SENSEVAL word sense disambiguation exercise and show that the bigram based decision tree approach is more accurate than the best SENSEVAL results for 19 of 36 words.
Human-annotated Summary
This paper presents a corpus-based approach to word sense disambiguation where a decision tree assigns a sense to an ambiguous word based on the bigrams that occur nearby.for this purpose the sense inventory of word sense has already been determined. This paper describes an approach where a decision tree is learned from some number of sentences where each instance of an ambiguous word has been manually annotated with a sense-tag that denotes the most appropriate sense for that context and for Building a Feature Set of Bigrams; two alternatives, the power divergence family and the Dice Coefficient were explored. this study utilizes the training and test data from the 1998 SENSEVAL evaluation of word sense disambiguation systems.The results of this approach are compared with those from the 1998 SENSEVAL word sense disambiguation exercise and show that the bigram based decision tree approach is more accurate than the best SENSEVAL results for 19 of 36 words.

Table 5: Comparison between the automatically generated and human-annotated summaries on the validation set. Paper identifier: N01-1011.

4.4 Testing of the Poli2Sum approach

The outcomes of the Poli2Sum approach on the CL-SciSumm-19 test set were submitted to the Shared Task and evaluated by the organizers against the ground truth. We submitted four different runs with slightly different configuration set-

tings for the summarization step (Task 2). The main characteristics of each run are summarized below.

1. *Run 1*: the regression algorithm uses the complete feature-set. The summaries are created considering a sentence-level limit in length.
2. *Run 2*: the regression algorithm uses the complete feature-set. The summaries are created considering a word-level limit in length.
3. *Run 3*: the regression algorithm does not use the title-similarity feature. The summaries are created as in *Run 1*.
4. *Run 4*: the regression algorithm does not use the title-similarity feature. The summaries created as in *Run 2*.

Table 6 reports the results achieved by the best performing system runs on the test set separately per task and evaluation metric.

Task	**Best Run**	**Metric**	**Result**
Task 1a	1-2-3-4	F1-score (sentence overlap)	0.092
	1-2-3-4	F1-score (Rouge-SU4)	0.034
Task 1b	1-2-3-4	F1-score (Classification)	0.229
Task 2	1	F1-Score Rouge-2 (Abstract)	0.364
	1	F1-Score Rouge-SU4 (Abstract)	0.196
	2	F1-Score Rouge-2 (Community)	0.209
	2	F1-Score Rouge-SU4 (Community)	0.112
	1	F1-Score Rouge-2 (Human)	0.218
	1	F1-Score Rouge-SU4 (Human)	0.144

Table 6: Shared Task results. Evaluation against the ground truth.

The Poli2Sum approach performed best (1st out of 104 runs) on Task 2 against the community summary (i..e, the target of the Poli2Sum training process), while it placed 31st and 72nd against the abstract and human summaries, respectively. Notice that, unlike the content of the human summary, the sentences of the abstract and community summaries are also part of the input data thus they can be selected by an extractive summarizer. Notice also that the abstract is self-contained, while the community summary may include also sentences from different sections of the paper. Therefore, the latter summary provides a broader description of the content of the paper.

The performance of Poli2Sum for the intermediate Tasks 1.a and 1.b are the same for all the submitted runs (i.e., 36th out of 98 submitted runs for *Task 1a*, 57th over 98 submitted runs for *Task 1b*).

5 Conclusions and future works

This paper describes the Poli2Sum system submitted to the CL-SciSumm Shared Task at SIGIR 2019 BIRNDL Workshop. The proposed approach relies on an ensemble of supervised models trained on a variety of textual and latent features. The features selected for the training phase are tailored to each task. The performance of the Poli2Sum approach was promising on Task 2, especially against the community summary, which is the target of the prediction process.

As future work, we plan to test the integration of deep learning architectures (e.g., BERT [5]) in the current architecture to solve similar research problems.

6 Acknowledgements

The research leading to these results has been partly funded by the SmartData@PoliTO center for Big Data and Machine Learning technologies.

References

1. Aburaed, Ahmed and Bravo, Alex and Chiruzzo, Luis and Saggion, Horacio: 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)
2. Bird, Steven and Klein, Ewan and Loper, Edward: Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc." (2009)
3. Breiman, Leo: Bagging predictors. Machine learning **24**(2), 123–140 (1996)
4. Chandrasekaran, M.K. and Yasunaga, M. and Radev, D. and Freitag, D. and Kan, M.-Y.: Overview and Results: CL-SciSumm SharedTask 2019. In: In Proceedings of the 4th Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2019) @ SIGIR 2019, Paris, France. (2019)
5. Devlin, Jacob and Chang, Ming-Wei and Lee, Kenton and Toutanova, Kristina: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
6. Elnaz Davoodi and Kanika Madan and Jia Gu: CLSciSumm Shared Task: On the Contribution of Similarity measure and Natural Language Processing Features for Citing Problem. In: BIRNDL@SIGIR. "CEUR" Workshop Proceedings, vol. 2132, pp. 96–101. CEUR-WS.org (2018)
7. Gaurav Baruah and Maheedhar Kolla: Klick Labs at CL-SciSumm 2018. In: BIRNDL@SIGIR. "CEUR" Workshop Proceedings, vol. 2132, pp. 134–141. CEUR-WS.org (2018)
8. Jaidka, Kokil and Yasunga, Michihiro and Chandrasekaran, Muthu and Radev, Dragomir and Kan, Min-Yen: The CL-SciSumm Shared Task 2018: Results and Key Insights. pp. 1–10 (07 2018)

9. Kumar Chandrasekaran, Muthu and Jaidka, Kokil and Mayr, Philipp: Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2018). In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. pp. 1415–1418. SIGIR '18, ACM, New York, NY, USA (2018). https://doi.org/10.1145/3209978.3210194, http://doi.acm.org/10.1145/3209978.3210194
10. Lei Li and Junqi Chi and Moye Chen and Zuying Huang and Yingqi Zhu and Xiangling Fu: CIST@CLSciSumm-18: Methods for Computational Linguistics Scientific Citation Linkage, Facet Classification and Summarization. In: BIRNDL@SIGIR. "CEUR" Workshop Proceedings, vol. 2132, pp. 84–95. CEUR-WS.org (2018)
11. Leskovec, Jure and Rajaraman, Anand and Ullman, Jeffrey David: Mining of Massive Datasets. Cambridge University Press, New York, NY, USA, 2nd edn. (2014)
12. Ley, Michael: The DBLP computer science bibliography: Evolution, research issues, perspectives. In: International symposium on string processing and information retrieval. pp. 1–10. Springer (2002)
13. Lin, Chin-Yew and Hovy, Eduard: Automatic evaluation of summaries using N-gram co-occurrence statistics. In: Proceedings of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1. pp. 71–78 (2003)
14. Luca Cagliero and Paolo Garza and Mohammad Reza Kavoosifar and Elena Baralis: Discovering cross-topic collaborations among researchers by exploiting weighted association rules. Scientometrics **116**(2), 1273–1301 (2018). https://doi.org/10.1007/s11192-018-2737-3, https://doi.org/10.1007/s11192-018-2737-3
15. Mikolov, Tomas and Sutskever, Ilya and Chen, Kai and Corrado, Greg S and Dean, Jeff: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
16. Pagliardini, Matteo and Gupta, Prakhar and Jaggi, Martin: Unsupervised learning of sentence embeddings using compositional n-gram features. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. vol. 1, pp. 528–540 (2018)
17. Pancheng Wang and Shasha Li and Ting Wang and Haifang Zhou and Jintao Tang: "NUDT" @ CLSciSumm-18. In: 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), Ann Arbor, USA, July 12, 2018. pp. 102–113 (2018)
18. Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E.: Scikit-learn: Machine Learning in "P"ython. Journal of Machine Learning Research **12**, 2825–2830 (2011)
19. Rosen-Zvi, Michal and Griffiths, Thomas and Steyvers, Mark and Smyth, Padhraic: The Author-topic Model for Authors and Documents. In: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence. pp. 487–494. UAI '04, AUAI Press, Arlington, Virginia, United States (2004), http://dl.acm.org/citation.cfm?id=1036843.1036902
20. Shutian Ma and Jin Xu and Chengzhi Zhang: Automatic identification of cited text spans: a multi-classifier approach over imbalanced dataset. Scientometrics **116**(2), 1303–1330 (2018)

21. Steven Ovadia: ResearchGate and Academia.edu: Academic Social Networks. Behavioral & Social Sciences Librarian **33**(3), 165–169 (2014). https://doi.org/10.1080/01639269.2014.934093
22. Sujatha Das Gollapalli and Cornelia Caragea: Extracting Keyphrases from Research Papers Using Citation Networks. In: AAAI (2014)
23. Tang, Jie and Zhang, Jing and Yao, Limin and Li, Juanzi and Zhang, Li and Su, Zhong: ArnetMiner: Extraction and Mining of Academic Social Networks. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 990–998. KDD '08, ACM, New York, NY, USA (2008). https://doi.org/10.1145/1401890.1402008, http://doi.acm.org/10.1145/1401890.1402008
24. Wu, Siyuan and U, Leong Hou and Bhowmick, Sourav S. and Gatterbauer, Wolfgang: Conflict of Interest Declaration and Detection System in Heterogeneous Networks. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. pp. 2383–2386. CIKM '17, ACM, New York, NY, USA (2017). https://doi.org/10.1145/3132847.3133134, http://doi.acm.org/10.1145/3132847.3133134