IRIT-IRIS at CL-SciSumm 2019: Matching Citances with their Intended Reference Text Spans from the Scientific Literature

Yoann Pitarch, Karen Pinel-Sauvagnat, Gilles Hubert,
Guillaume Cabanac, Ophélie Fraisier-Vannier

{yoann.pitarch, karen.sauvagnat, gilles.hubert, guillaume.cabanac, ophelie.fraisier}@irit.fr,
Université de Toulouse, IRIT UMR 5505 CNRS,
118 route de Narbonne, F-31062 Toulouse cedex 9

Abstract. The CL-SCisumm track provides a framework to evaluate systems summarising scientific papers. It includes datasets and metrics provided by the organisers. The track comprises three tasks: (1a) identifying the spans of text in the referred document reflecting citing text spans (i.e., citances), (1b) classifying discourse facets of the cited text spans, and (2) generating a short structured summary. For the 2019 edition, we focused our work on the task 1a. This report presents our proposed approach for this task. We submitted 15 runs corresponding to different configurations of the parameters involved in our approach.

1 Our Hypothesis

As a reminder the aim of task 1a [1] on which we focused our efforts is the following: Given a *Citation Paper (CP)* and a precise source of citation in CP, find the target(s) of citation at sentence level in the *Reference Paper RP*. The source of citation in CP is called a *Citance* while the target of citation in RP is called a *Reference Text Span*. Both Citance and Reference Text span can be composed of one or more sentences (respectively in CP and RP).

We based our approach for the task 1a on the key insight that all the sentences of a reference paper can be selected as target of a citation, i.e., as reference text span.

Following this key insight, we propose to define an approach that first attempts to identify candidate target sentences based on features characterising the sentences usually targeted by citations. A first issue is thus to design features to identify candidate sentences within RPs. A second issue is then to estimate, for each sentence, the probability of being a possible target. The final target sentences of a citance have then to be selected in the set of possible targets according to an appropriate strategy.

2 Methods

Based on our hypothesis presented in the previous section we defined a two-step approach described in the following sections.

2.1 Step 1: Identification of Candidate Sentences

To identify candidate sentences, we converted this problem into a standard binary classification problem. Considering a training dataset $\mathcal{D}_T$, a sentence belongs to the positive class if it is targeted by at least one CP. Otherwise, the sentence belongs to the negative class. We then

designed features in order to characterise candidate sentences of RPs related to four main categories: bibliographic features, conceptual features, positional features, and features based on word distribution. The features used in our approach are presented in Tab. 1.

For features based on the word distribution, the Iramuteq software [6] was used to identify the significant words in each class (positive and negative class), based on the χ^2 indicator. Features $\text{f14}_i - \text{f19}_i$ (resp. $\text{f20}_i - \text{f23}_i$) retained the top i over- (resp. under-) represented terms ($i \in \{5, 10, 20, 30, 40, 50\}$).

Second, we applied a machine learning approach, i.e., XGBoost [2], to learn the model that best estimates the probability of a RP sentence to be a good target sentence, i.e., to belong to the positive class. In the standard binary classification settings, a sentence would be predicted as a target sentence if the estimated probability is strictly greater than 0.5. Since our objective is slightly different, i.e., filtering out noisy sentences, we thus introduced a threshold λ to filter out sentences that are not likely to be target sentences. Specifically, given a sentence, if its estimated probability is strictly lower than λ, this sentence is no longer considered in Step 2.

2.2 Step 2: Computation of Sentence–Sentence Similarities

Here, we aimed to find the most similar sentences in the RP for each citance. We represented sentences as vectors whose values stem from applying tf·idf-based methods or applying embedding-based methods such as Word2vec [5].

Simple tf·idf. We first decided to evaluate a very simple representation of candidate sentences and citances, based on the well-known Vector Space Model. The vector representation of a sentence is thus based on *tf·idf*. We considered two different vocabularies: one composed of all terms in RPs, and one after performing a POS-tagging using the Python library Spacy [3], and keeping only nouns, adjectives, and verbs. In both cases, *idf* was evaluated at sentence level: as the total number of sentences in RP divided by the number of sentences in RP containing the considered term.

Embedding-based Methods. For embedding-based word representation, we trained a deep learning model on our in-house WoS-CS corpus. This consists in textual data pulled from the *Web of Science*, covering the 1.6 million abstracts (of length 140+ characters) of all 2005–2018 articles and proceeding papers published in venues listed in the following four fields of the 'Computer Science' subject unit: Information Systems, Artificial Intelligence, Interdisciplinary Applications, Theory & Methods. Prior to training the model, we produced a lowercased, diacritic- and punctuation-free version of WoS-CS. We then fed it to Word2vec [5] set up with continuous skip-gram architecture, which produced a set of 200-dimension vectors: one vector for each word of the corpus. Each vector encodes a representation of the underlying word as it appeared in its context.

Given a sentence s, we averaged the embedding vectors of words in s. To avoid considering non-informative words i in the aggregation process, we first performed a POS-tagging using the Python library Spacy. We then averaged vectors of nouns, adjectives, and verbs only. Note that this POS filtering is not performed for all of our runs as specified in Sect. 4.

Matching between Citances and Candidate Sentences in the RP. Once the vector representation of sentences has been calculated, we then computed the cosine similarity for

Table 1. Description of the features we designed. They are evaluated for each sentence of RPs, i.e. for each candidate sentence. We used in our experiments $i \in \{5, 10, 20, 30, 40, 50\}$. Polarity reflects the hypothesis whether the feature is a positive or a negative one, i.e. expected to discriminate a sentence in the positive or negative class.

Name	Description	Polarity
	BIBLIOGRAPHIC	
f1	Presence of a bibliographic reference	Pos
	CONCEPTUAL (within RP)	
f2	Number of common words between RP title and RP sentence	Pos
f3	Cosinus between RP title and RP sentence embeddings	Pos
f4	Cosinus between RP title and RP sentence embeddings (weighted by TF-IDF)	Pos
	CONCEPTUAL (between all Citances and RP)	
f5	Max number of words in common with a Citance	Pos
f6	Max cosinus with a Citance	Pos
f7	Max cosinus with a Citance considering embeddings	Pos
	POSITIONAL	
f8	Sentence in Acknowlegments section	Neg
f9	Sentence in References section	Neg
f10	Normalized sentence position in the paper – $^{sentencePosition}/_{numberOfSentences}$	Pos
f11	Normalized sentence position in the corresponding section ($ssid$) – 0 for titles, $^{ssid}/_{\max(ssid)}$ for other sentences	Pos
f12	Normalized section number – 0 for titles, $^{sectionNumber}/_{\max(sectionNumber)}$ for other sections (the Acknowledgements section was numeroted by following the previous sections)	Pos
f13	Sentence's section label, compared to a predefined set of labels ({abstract, introduction, model, method, results, experiments, conclusion, rw, others}). The label is assigned regarding the keywords found in the title of the section.	Pos
	OVER-REPRESENTED WORDS IN T	
f14	Presence of the most over-represented word	Pos
$f15_i$	Presence of at least one of the i most over-represented words	Pos
$f16_i$	Number of the i most over-represented words present	Pos
f17	Presence of the most over-represented word in the section title	Pos
$f18_i$	Presence of at least one of the i most over-represented words in the section title	Pos
$f19_i$	Number of the i most over-represented words present in the section title	Pos
	UNDER-REPRESENTED WORDS IN T	
$f20_i$	Presence of at least one of the i most under-represented words	Neg
$f21_i$	Number of the i most under-represented words present	Neg
$f22_i$	Presence of at least one of the i most under-represented words in the section title	Neg
$f23_i$	Number of the i most under-represented words present in the section title	Neg

each pair of (candidate target sentence, citance) and ranked the candidate target sentence in decreasing order of similarity. A maximum of n target sentences were finally selected with a similarity greater or equal to a threshold α.

3 Preliminary Experiments

We carried out a series of preliminary experiments aiming to draw a set of effective candidate configurations of our system. We used the test set and the evaluation framework of the 2018 CL-SciSumm edition as well as the training set of 2019 to compare various configurations of our system according to their F1-scores for sentence overlap on Task 1a.

One experiment intended to evaluate the interest of using automatically vs. manually annotated documents for training. As shown in Tab. 2, we experimented our system firstly training on the Training 2019 set and testing on the Training 2018 set, and secondly training on the Training 2018 set and testing on the Training 2018 with cross-validation. The comparison of the obtained evaluations did not lean towards a training on the automatically annotated documents of the Training 2019 set. As a consequence, we built runs using three training sets for the first step: Training 2018, Training 2019, and Training 2018 + Training 2019.

Table 2. Preliminary experiments to evaluate the interest of training on automatically vs. manually annotated documents

Training set	Test set
Training 2019 (1 000 automatically annotated documents)	Training 2018
Training 2018 (40 manually annotated documents)	Training 2018 (cross-validation)

Other preliminary experiments were conducted in order to identify various configurations of the components (embeddings generation, with or without POS-tagging) and the parameters (λ, n, α) performing well in the context of the 2018 edition [4].

4 Submitted Runs

We submitted 15 runs based on our approach to address the subtask 1a. The submitted runs correspond to different configurations of the components and parameters involved in our approach (Tab. 3).

On the one hand, the first varying parameter was the training set used for the first step among Training 2018, Training 2019, and Training 2018 + Training 2019 sets as mentioned in the previous section. The other parameters were the threshold λ used to select the sentences to retain as candidate for being target of citances, the threshold α used to select the sentences best matching the citances, and finally the maximum number n of selected target sentences. On the other hand, the varying components were generating word embeddings or not for the vectors representing sentences and applying POS tagging or not.

Table 3. Submitted runs with the applied parameters. The column labelled Emb. indicates if embeddings are computed or not. The column labelled POS indicates if POS tagging is applied or not.

Run name	Training set	λ	Emb.	POS	α	n
WithoutEmbPOS_Training20182019_Test2019_3_0.10	2018+2019	0.10	No	Yes	0.00	3
WithoutEmbPOS_Training2018_Test2019_3_0.10	2018	0.10	No	Yes	0.00	3
WithoutEmbPOS_Training2019_Test2019_3_0.10	2019	0.10	No	Yes	0.00	3
WithoutEmbTopsimPOS_Training20182019_Test2019_0.15_5_0.05	2018+2019	0.05	No	Yes	0.15	5
WithoutEmbTopsimPOS_Training2018_Test2019_0.15_5_0.05	2018	0.05	No	Yes	0.15	5
WithoutEmbTopsimPOS_Training2019_Test2019_0.15_5_0.05	2019	0.05	No	Yes	0.15	5
WithoutEmbTopsim_Training20182019_Test2019_0.15_5_0.05	2018+2019	0.05	No	No	0.15	5
WithoutEmbTopsim_Training2018_Test2019_0.15_5_0.05	2018	0.05	No	No	0.15	5
WithoutEmbTopsim_Training2019_Test2019_0.15_5_0.05	2019	0.05	No	No	0.15	5
WithoutEmb_Training20182019_Test2019_3_0.10	2018+2019	0.10	No	No	0.00	3
WithoutEmb_Training2018_Test2019_3_0.10	2018	0.10	No	No	0.00	3
WithoutEmb_Training2019_Test2019_3_0.10	2019	0.10	No	No	0.00	3
unweightedPOS_W2v_Training20182019_Test2019_3_0.05	2018+2019	0.05	Yes	Yes	0.00	3
unweightedPOS_W2v_Training2018_Test2019_3_0.05	2018	0.05	Yes	Yes	0.00	3
unweightedPOS_W2v_Training2019_Test2019_3_0.05	2019	0.05	Yes	Yes	0.00	3

5 Evaluation Results

The official evaluation results for our submitted runs (see Tab. 3) are reported in Table 4.

Table 4. Evaluation results for Task 1a for the submitted runs. F1-SO refers to Task1A: Sentence Overlap (F1) and F1-RO refers to Task1A: ROUGE-SU4 (F1).

Run name	F1-SO	F1-RO
WithoutEmbPOS_Training20182019_Test2019_3_0.10	0.089	0.065
WithoutEmbPOS_Training2018_Test2019_3_0.10	0.089	0.065
WithoutEmbPOS_Training2019_Test2019_3_0.10	0.089	0.065
WithoutEmbTopsimPOS_Training20182019_Test2019_0.15_5_0.05	0.088	0.044
WithoutEmbTopsimPOS_Training2018_Test2019_0.15_5_0.05	0.088	0.044
WithoutEmbTopsimPOS_Training2019_Test2019_0.15_5_0.05	0.088	0.044
WithoutEmbTopsim_Training20182019_Test2019_0.15_5_0.05	0.090	0.044
WithoutEmbTopsim_Training2018_Test2019_0.15_5_0.05	0.090	0.044
WithoutEmbTopsim_Training2019_Test2019_0.15_5_0.05	0.090	0.044
WithoutEmb_Training20182019_Test2019_3_0.10	0.097	0.071
WithoutEmb_Training2018_Test2019_3_0.10	0.097	0.071
WithoutEmb_Training2019_Test2019_3_0.10	0.097	0.071
unweightedPOS_W2v_Training20182019_Test2019_3_0.05	0.076	0.047
unweightedPOS_W2v_Training2018_Test2019_3_0.05	0.076	0.045
unweightedPOS_W2v_Training2019_Test2019_3_0.05	0.076	0.045

A first conclusion that can be drawn is that the simple tf-idf representation outperforms the one based on embeddings. Best results are obtained when returning 3 sentences per reference text span.

Surprisingly however, results are strictly simlilar whatever the training set used. Further investigations are needed to understand these results. We are also waiting for the ground

truth to perform a failure analysis and deeply investigate on the effectiveness of each step of our approach.

References

1. Chandrasekaran, M., Yasunaga, M., Radev, D., Freitag, D., Kan, M.Y.: Overview and Results: CL-SciSumm SharedTask 2019. In: Proceedings of the 4th Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2019) @ SIGIR 2019 (2019)
2. Chen, T., Guestrin, C.: XGBoost: A Scalable Tree Boosting System. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 785–794. KDD '16, ACM, New York, NY, USA (2016)
3. Honnibal, M., Montani, I.: spaCy : Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear (2017)
4. Jaidka, K., Yasunaga, M., Chandrasekaran, M.K., Radev, D.R., Kan, M.: The CL-SciSumm Shared Task 2018: Results and Key Insights. In: Mayr, P., Chandrasekaran, M.K., Jaidka, K. (eds.) 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. CEUR Workshop Proceedings, vol. 2132, pp. 74–83. CEUR-WS.org (2018)
5. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. pp. 3111–3119. NIPS'13, Curran Associates Inc., USA (2013)
6. Ratinaud, P.: IRaMuTeQ : Interface de R pour les Analyses Multidimensionnelles de Textes et de Questionnaires (2009), `http://www.iramuteq.org`