RALI System Description for CL-SciSumm 2016 Shared Task

Bruno Malenfant[1] and Guy Lapalme[2]

[1] Université de Montréal, CP 6128, Succ Centre-Ville, Montréal, Québec, Canada, H3C 3J3, malenfab@iro.umontreal.ca
[2] Université de Montréal, CP 6128, Succ Centre-Ville, Montréal, Québec, Canada, H3C 3J3, lapalme@iro.umontreal.ca

Abstract. We present our approach to the CL-SciSumm 2016 shared task. We propose a technique to determine the discourse role of a sentence. We differentiate between words linked to the topic of the paper and the ones that link to the facet of the scientific discourse. Using that information, histograms are built over the training data to infer a facet for each sentence of the paper (*result*, *method*, *aim*, *implication* and *hypothesis*). This helps us identify the sentences best representing a citation of the same facet. We use this information to build a structured summary of the paper as an HTML page.

1 Introduction

One's task in research is to read scientific papers to be able to compare them, to identify new problems, to position a work within the current literature and to elaborate new research propositions [8].

This implies reading many papers before finding the ones we are looking for. With the growing amount of publications, this task is getting harder. It is becoming important to have a fast way of determining the utility of a paper for our needs. A first solution is to use web sites such as *CiteSeer, arXiv, Google Scholar* and *Microsoft Academic Search* that provide cross reference citations to papers. Another approach is automatic summarization of a group of scientific papers dealing with the subject.

This year's CL-SciSumm competition for summarization of computational linguistics papers proposes a community approach to summarization; it is based on the assumption that citances, the set of citation sentences to a reference paper, can be used as a measure of its impact. This task implies identifying the text a citance refers to in the reference paper and a facet (*aim*, *result*, *method*, *implication* and *hypothesis*) for the referred text.

We are building a system that given a topic, generates a survey of the topic from a set of papers. That system uses citations as the primary source of information for building an annotated summary. Our system must be able to identify the purpose/polarity/facet of a citation. to direct the reader towards the more relevant information. The summary is built by selecting sentences from the cited paper and the citations. This process uses a similarity function between sentences. The resulting summaries are presented in HTML format with their annotations and links to the original paper. The only task that is not

performed by our system is finding the text referred to by the citation. We intend to use the information already found by our system (facet of citations and sentences) to complete that task.

We had already some experience in dealing with scientific papers and their references, having participated to Task2 of the Semantic Publishing Challenge of ESWC-2014 (Extended Semantic Web Conference) on the extraction and characterization of citations. A short review of previous work follows in Sect. 2. We will summarize the task in Sect. 3 and the techniques for extracting information in Sect. 4. Finally, Sect. 5 will show our results.

2 Previous Work

There has been a growing attention towards the information carried by citations and their surrounding sentences (citances). These contain information useful for rhetorical classification [18], technical surveys [14] and emphasize the impact of papers [12]. Qazvinian [16] and Elkiss [6] showed that citations provide information not present in the abstract.

Since the first works of Luhn [11] and Edmundson [5] many researchers have developed methods for finding the most relevant sentences of papers to produce abstracts and summaries. Many metrics have been introduced to measure the relevance of parts of text, either using special purpose formulas [21] or using learned weights [10]. The hypothesis for CL-SciSumm task is that important sentences can be pointed out by other papers : a citation indicates a paper considered important by the author of the citing paper.

Another domain for study over scientific papers is the classification of their sentences. Teufel [19] identified the rhetorical status of sentences using Bayes classifier.

To find citations inside a paper, we need to analyse the references section. Dominique Besagni et al. [1] developed a method using pattern recognition to extract fields from the references while Brett Powley and Robert Dale [15] looked citations and references simultaneously using informations from one task to help complete the second task.

3 Task Description

For this year competition we were given 30 topics, 10 for training, 10 for tuning and 10 for testing [9]. Each topic is composed of a Reference Paper (RP) and some Citing Papers (CPs). The citing papers contain citations pointing to the RP. An annotation file is given for each topic. That file contains information about each citation, the citation marker and the citance.

There are two mandatory tasks (Task 1A and Task 1B) and an optional task (Task 2)[3].

Task 1A : Find the part of the RP that is indicated with each citance. This will be called the *referenced text*.

[3] `http://wing.comp.nus.edu.sg/cl-scisumm2016/`

Task 1B : Once the referenced text is identified, we need to attribute a facet to it. A facet is one of these : *result*, *method*, *aim*, *implication* and *hypothesis*.

Task 2 : Building a summary for the RP using the referenced text identified in Task 1A.

Both the training and the developing set of topics contain expected results for these tasks. The next section will describe how our system performs on the test set.

4 Our Approach

For the first task, we have to find the referenced text and its facet. We hypothesized that the referenced text should be sentences sharing the same facet as the citance. We use that fact to reduce the set of sentences to choose from for the reference. This is why we execute Task 1B on all the sentences of the RP and all the citances prior to Task 1A. We now present how we determine the facet of a citance, then the facet for sentences in the RP and finally the referenced text.

4.1 Task 1B : Facet Identification

Our goal is to be able to use our system for papers from different domains, without having to train them again. Toward that objective, our system only uses words that are not domain specific. Patrick Drouin [3, 4] compiled such a list of words in his *Transdisciplinary scientific lexicon* (TSL). This lexicon comprises 1627 words such as *acceptance, gather, newly, severe*... We will denote the set of words from the lexicon using $w \in L$.

We trained two systems, one to attribute a facet to sentences in the RP and one to attribute a facet to citances.

We determine the word distribution for each facet using an histogram. We only use words appearing in the TSL. This computation yielded a sum of each words present in all referenced text for each facet. The facet with the highest score is chosen for that sentence.

For training our system, we extract the reference sentences from each annotation with their assigned facet. Each sentence is tokenized using the NLTK library in Python. Only words from the TSL are kept. Our dataset consists of pairs of list of words with a facet : $D = [(\mathrm{ws}_i, f_i)]$.

We build a profile (h_f) for each facet using a histogram. For each word in the lexicon, we compute the number of times it appears in sentence paired with the a specific facet.

$$h_f(w, D) = \sum [\mathrm{cnt}(w, \mathrm{ws}_i) \mid (\mathrm{ws}_i, f) \in D]$$
$$\mathrm{cnt}(w, \mathrm{ws}_i) = \sum [1 \mid w \in \mathrm{ws}_i]$$

When a word appears more then once in a sentence, all its occurrences are counted.

Once the histogram is built, we use it to find the facet of new sentence. First, we extract the words that are part of the lexicon from the sentence, yielding the list of words

p. Then a score s_f for each facet is computed by adding the profile of each word for that facet. The facet that scored the highest value is assigned to the new sentence.

$$s_f(p, D) = \sum_{w \in p} h_f(w, D)$$

Looking closely at the results for the profile, we saw that some words have a negative effect on finding the facet. To find a better sublist of words to use within the TSL, we used a genetic algorithm that uses a population of lists of words.

A genetic algorithm starts with an initial population (set of possible solutions) and tries to find better solutions by applying small changes to existing solutions. In our case, a solution is a subset of words L_i of L. The initial population is built using random subsets.

To build the next generation, we use three different techniques :

1. Adding a random word to an existing solution : $L'_i = L_i + \{w\}$ where $w \in (L - L_i)$.
2. Removing a random word from an existing solution : $L'_i = L_i - \{w\}$ where $w \in L_i$.
3. Combining two subsets of existing solutions : $L'_i = L_j \cup L_k$.

Once enough solutions are built for the new population, each solution is tested using cross-validation with the histogram. The list that performed best in the task is kept for the next generation. We use the same technique over the dataset consisting of the citance texts and their facets.

4.2 Task 1A : Finding the Sentences Referred to by Citances

Having determined the facet of sentences in both the RP and citances, we are now ready to assign referenced text to citances from the CPs. Our hypothesis is that a citance should have the same facet as the text it refers to. We extract Q_f the subset of sentences from the RP that have the same facet f as a citance c_i. To choose the sentence of RP referred to by the citance, we look for the sentence from Q_f that is the most similar with the citance c_i.

$$\mathrm{sim}_{mcs}(P_1, P_2) = \frac{1}{2}\left(\mathrm{hs}(P_1, P_2) + \mathrm{hs}(P_2, P_1)\right) \quad (1)$$

$$\mathrm{hs}(P_i, P_j) = \frac{\sum_{w \in P_i} \mathrm{ms}(w, P_j) \times \mathrm{idf}_w}{\sum_{w \in P_i} \mathrm{idf}_w} \quad (2)$$

$$\mathrm{ms}(w, P_j) = \max_{v \in P_j} \mathrm{sim}_{wup}(w, v) \quad (3)$$

We use the similarity function sim_{mcs} defined by Mihalcea, Corley and Strapparava [13]. This similarity function between sentences P_1 and P_2 (Equation 1) averages two values, the similarity from P_1 to P_2, and the similarity from P_2 to P_1. The similarity from one sentence P_i to the other P_j is computed by first pairing each word from the first sentence $w \in P_i$ with a word in the second one $v \in P_j$. A word is paired

with the one that is the most similar to it (Equation 3). For each pair (w, v), the value of the similarity is weighted by the Inverse Document Frequency of the first word idf_w (Equation 2). The average of these weighted similarity values is computed to yield the similarity between P_i and P_j. We use only words that are Noun, Verb, Adjective and Adverb for this comparison. The `POS_tagger` of **NLTK** was used to assert the tag of each word. Since we believe that the domain of the paper is important to compute that similarity, we use all words, not only the ones that are part of the TSL.

Mihalcea et al. [13] reported that within the set of possible metrics to compare words, the one proposed by Zhibiao Wu et Martha Palmer [20] yielded good result (denoted sim_{wup}). This metric is also available with the **NLTK** package. To use that metric, we transform each word into their synonym group synset using **WordNet**. The IDF was computed for each synset. The computation was done over the set of all the documents contained in the ACL Anthology Network[4].

4.3 Task 2 : Summarization

Multiple source summarization adds three problems [17] :

1. Redundancy : a paper will often be cited for the same reason over and over, resulting in many citances having the same subject.
2. Identifying important differences between sources : our goal will be to find those citances/references that bring new information and important information to the summary.
3. Coherence : since sentences come from many sources, we want to ensure that the summary forms an unified whole.

For Task 2, we choose to use the *Maximal Marginal Relevance* (MMR) proposed by Jaime G. Carbonell et Jade Goldstein [2]. Their technique is presented in Equation 4, in which R is the list of possible sentences and V is the summary. They propose to use the title of the research paper as the starting query Q.

$$\underset{s_i \in R \setminus V}{\arg\max} \left[\lambda \ \mathrm{sim}_{mcs}(s_i, Q) - (1 - \lambda) \max_{s_j \in V} \mathrm{sim}_{mcs}(s_i, s_j) \right] \tag{4}$$

At each iteration, their algorithm adds a sentence s_i to V. Sentences are choosen so that they bring new information to the summary (Points 1 and 2) and it must have a certain amount of similarity with the query (Point 2). λ must be adjusted to balance between adding a sentence very similar to the query and a sentence very different from the ones already in the summary V. We use the same metric (sim_{mcs}) as for task 1A to compare sentences.

We divided the summarization process in two steps : adding sentences from the citance ($R = \mathrm{CT}$) and adding sentences from the paper ($R = \mathrm{RP}$). In the first step, the algorithm chooses sentences from the set of citances until it reaches 150 words. For that part, we use $\lambda = 0.3$ to give priority to similarity with the query, trying to remove meaningless citances. Since citances have been identified as bringing new information

[4] `http://clair.eecs.umich.edu/aan/index.php`

not present in the original paper, we believe it is important to keep them in the summary. Then, the summary is completed (to 250 words) using sentences choosen from the RP. Here, we use $\lambda = 0.7$. Since sentences are choosen in the RP, most of them are about the same subject, we want to give priority to sentences that are more different.

The summary is built in an XML format. Each sentence is identified with its position (the `id` of the paper it was extracted from, the `sid` and `ssid` attributes inside the XML source files). The citances contain the `id` of the referred paper. This information will enable to point a reader towards the corresponding paper.

To help analyse the summaries, our software builds an HTML page containing the extracted information (see Fig. 1).

5 Evaluation

5.1 Task 1

We present our results for facet attribution to citance and reference text. The set of data we receive is divided in two : the training set contains 197 sentences distributed over the citances and 247 sentences over the reference text; the development set contains 273 sentences distributed over the citances and 330 sentences over the reference text. We first train our system using the training data (**T**) and then we retrained it using both set training and developing set together (**TD**). In each case, we test the result over both sets. We show the result for simple training of the histogram and for the training using the genetic algorithm (**gen_T**) to select the list of words to consider. We also trained our histogram without limiting to the words in the TSL for comparison purpose.

For the genetic algorithm, we let it run over 25 generations. Each generation started with 1 000 lists of words. 9 000 lists are added using the proposed mutations, bringing the number of lists to 10 000.

Table 1. Success rate for attributing facet to citances.

	Tested on		
Trained on	**Train**	**Dev**	**Train + Dev**
T no TSL	47%	61%	59%
T	65%	52%	57%
TD no TSL	56%	61%	59%
TD	61%	57%	58%
gen_T	74%	43%	55%

The result of these experiments are presented in Table 1 and Table 2. We see that, using the training set **T** gives good result on itself but lower result when we apply it on the development set. After training with both set **TD** (Test + Development), the result over the development set raises at the expense of the result for the training set. For citance, the genetic algorithm yields better result over the training set only. It does not help to get better histograms. Considering that fact, we ask ourselves if it is possible to

Table 2. Success rate for attributing facet to references text.

	Tested on		
Trained on	**Train**	**Dev**	**Train + Dev**
T no TSL	60%	60%	60%
T	74%	46%	57%
TD no TSL	60%	61%	61%
TD	70%	59%	64%
gen_T	76%	35%	51%

obtain better results using histograms, or if we have reached the limit of that technique? Limiting our choice of words to the TSL did not give lower results. It is to be tested if the histograms built with the TSL will perform better in another domain than computational linguistics.

Once we had identified the facets, we ran our script for finding the reference text. It was able to reach an F1 score of 0.095 over the training set and 0.052 over the development set (table 3). We reduced the search space for the referred text using the facet of the citance. Since the identification of the facet is not perfect, this reduction might remove a sentence we are looking for. In the future, we have to test our approach with all sentences, instead of the reduced set, to see if this reduction of space causes a problem more than helps the solution.

Table 3. F1 scores for finding the reference text.

	Train	**Dev**
F1	0.095	0.052

5.2 Task 2

Figure 1 shows the HTML interface we have generated for showing the result of our system. It allows for selecting different topics. The top of the page lets us choose between the different topics that were summarised. Each topic will present, on the left side, the text of each CPs and RP. The sentences have been divided and citance identified. The right side contains the different summaries that our software builds (using different values of λ) and the gold standard summary. Each paper links to its pdf version on the ACL Anthology[5].

On the left side of the top part of the figure we see the RP divided in sentences. On the right side, there is a summary built by choosing five sentences from the set of citances using a λ of 0.3. These sentences where selected to be as different as possible by the MMR algorithm. The bottom screen shoot (Fig. 1) presents one of the CP on the left. The citance and citation are colored to be easy to identify. The third sentence from the top was selected by the algorithm for the summaries.

[5] `http://aclanthology.info/`

6 Conclusion

We presented the use of distinguishing between topic and non-topic (TSL) words for determining the facet of sentences in a paper. This technique is useful because it lets our system work on paper in a domain independent way. We obtained good results with a simple histogram. We still have to test our histogram over other domains, to see if they also yield good results. Our experiments with a genetic algorithm to refine the list of used words did not show any improvement.

We presented our interface for browsing the results of our system. That interface presents RP, CPs and summaries with links to the original paper. This interface helps the reader browse through a topic.

References

1. Dominique Besagni, Abdel Belaïd, and Nelly Benet : A Segmentation Method for Bibliographic References by Contextual Tagging of Fields. *ICDAR '03 Proceedings of the Seventh International Conference on Document Analysis and Recognition*, 1:384–388 (2003)
2. Jaime G. Carbonell, and Jade Goldstein : The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries. *Research and Development in Information Retrieval - SIGIR*, 335–336 (1998)
3. Patrick Drouin : Extracting a Bilingual Transdisciplinary Scientific Lexicon. *Proceedings of eLexicography in the 21st Century : New Challenges, New Applications*. Presses universitaires de Louvain, Louvain-la-Neuve, 7:43–54 (2010)
4. Patrick Drouin : From a Bilingual Transdisciplinary Scientific Lexicon to Bilingual Transdisciplinary Scientific Colloations. *Proceedings of the 14th EURALEX International Congress*. Fryske Akademy, Leeuwarden/Ljouwert, Pays-Bas, 296–305 (2010)
5. Harold P. Edmundson : New Methods in Automatic Extracting. *Journal of the ACM (JACM)*, 16(2):264–285 (1969)
6. Aaron Elkiss, Siwei Shen, Anthony Fader, Günes Erkan, David J. States, and Dragomir R. Radev : Blind Men and Elephants: What Do Citation Summaries Tell Us About a Research Article?. *Journal of the American Society for Information Science and Technology - JASIS*, 59(1):51–62 (2008)
7. C. Lee Giles and Kurt D. Bollacker and Steve Lawrence : CiteSeer: an Automatic Citation Indexing System. *Proceedings of the Third ACM Conference on Digital Libraries*, 89–98 (1998)
8. Kokil Jaidka, Christopher S.G. Khoo, Jin-Cheon Na, and Wee Kim Wee : Deconstructing Human Literature Reviews – A Framework for Multi-Document Summarization. *Proceedings of the 14th European Workshop on Natural Language Generation*, 125–135 (2013)
9. Kokil Jaidka, Muthu Kumar Chandrasekaran, Sajal Rustagi, and Min-Yen Kan : Overview of the 2nd Computational Linguistics Scientific Document Summarization Shared Task (CL-SciSumm 2016). To appear in the *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2016)*, Newark, New Jersey, USA. (2016)
10. Julian Kupiec, Jan O. Pedersen, and Francine Chen : A Trainable Document Summarizer. *Research and Development in Information Retrieval - SIGIR*, 68–73 (1995)
11. Hans P. Luhn : The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development - IBMRD*, 2(2):159–165 (1958)
12. Qiaozhu Mei, and ChengXiang Zhai : Generating Impact-Based Summaries for Scientific Literature. *Meeting of the Association for Computational Linguistics - ACL*, 816–824 (2008)

13. Rada Mihalcea, Courtney Corley, and Carlo Strapparava : Corpus-based and Knowledge-based Measures of Text Semantic Similarity. *AAAI*, 6:775–780 (2008)
14. Saif Mohammad, Bonnie J. Dorr, Melissa Egan, Ahmed Hassan, Pradeep Muthukrishnan, Vahed Qazvinian, Dragomir R. Radev, and David M. Zajic : Using Citations to Generate Surveys of Scientific Paradigms. *North American Chapter of the Association for Computational Linguistics - NAACL*, 584–592 (2009)
15. Brett Powley, and Robert Dale : Evidence-based Information Extraction for High Accuracy Citation and Author Name Identification. *RIAO '07 Large Scale Semantic Access to Content*, 618–632 (2007)
16. Vahed Qazvinian, Dragomir R. Radev, Saif Mohammad, Bonnie J. Dorr, David M. Zajic, M. Whidby, and T. Moon : Generating Extractive Summaries of Scientific Paradigms. *Journal of Artificial Intelligence Research*, 46:165–201 (2013)
17. Dragomir R. Radev, Eduard Hovy, and Kathleen McKeown : Introduction to the Special Issue on Summarization. *Computational Linguistics - Summarization*, 28(4):399–408 (2002)
18. Advaith Siddharthan, and Simone Teufel : Whose Idea Was This, and Why Does it Matter? Attributing Scientific Work to Citations. *North American Chapter of the Association for Computational Linguistics - NAACL*, 316–323 (2007)
19. Simone Teufel, and Marc Moens : Summarizing Scientific Articles: Experiments with Relevance and Rhetorical Status. *Computational Linguistics - COLI*, 28(4):409–445 (2002)
20. Zhibiao Wu, and Martha Palmer : Verbs Semantics and Lexical Selection. *ACL '94 Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, 133–138 (1994)
21. Peter N. Yianilos, and Kirk G. Kanzelberger : The LikeIt Intelligent String Comparison Facility. *NEC Research Institute*. (1997)

Fig. 1. Screen shots of the HTML interface. The top part shows the RP and the corresponding summary. The bottom part shows a CP in which we see sentences from the CP where chosen in the summary.