Rali System Description For TAC 2014 Biomedical Summerization Track

Bruno Malenfant
Université de Montréal
CP 6128, Succ Centre-Ville
Montréal, Québec, Canada, H3C 3J3
malenfab@iro.umontreal.ca

Guy Lapalme
Université de Montréal
CP 6128, Succ Centre-Ville
Montréal, Québec, Canada, H3C 3J3
lapalme@iro.umontreal.ca

Abstract

We present our solution to the 2014 Biomedical Summerization Track. We propose a technique to determine the discourse role of a sentence. We differentiate words linked to the topic of the paper from the ones that link to the facet of the scientific discourse. Using that information, simple histograms are built over training data to infer a facet for each sentence of the paper (result, method, implication, hypothesis and discussion). This helps us isolate the sentences best representing a citation of the same facet.

1 Introduction

One's task in research is to read scientific paper to be able to compare them, to identify new problems, to position a work within the current literature and to elaborate new research propositions(Jaidka et al., 2013). 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 scientific paper. This year's TAC competition for summarization of biology 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 a facet (discussion, result, method, implication and hypothesis) for each citance and the text it refers to in the reference paper. To solve this task, we propose to build sets of words that identify these facets. Our method takes into account words that are present in any scientific paper without consideration for the subject. Patrick Drouin (2010a; 2010a) developed such a set, the *Lexique scientifique transdisciplinaire* (LST).

To assess the facet and find the reference text, we have tested with different subsets of the LST with the hypothesis that words from the LST can help identify the facets of sentences in a paper and words outside of the LST represent the topics. Our experiments show that indeed the words from the LST are good indicators for identifying the facet of a passage.

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 Section 2. We will present the preprocessing steps we use over the data in Section 3 and the techniques for extracting information in Section 4. Finally, Section 5 will show our results.

2 Previous Work

There has been a growing attention toward the information carried by citations and their surrounding sentences (citance). These contain information useful for rhetorical classification (Advaith Siddharthan and Simone Teufel, 2007), for technical

surveys (Saif Mohammad et al., 2009) and for emphasizing the impact of papers (Qiaozhu Mei and ChengXiang Zhai, 2008). Qazvinian (2013) and Elkiss (2008) showed that citations provide information not present in the author's abstract.

Since the first works of Luhn (1958) and Edmundson (1969) 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 (Peter N. Yianilos and Kirk G. Kanzelberger, 1997) or using learned weights (Julian Kupiec et al., 1995). The hypothesis for TAC 2014 task is that important sentences can be pointed out by other paper : the citation indicates a section of paper that was considered important by a reader.

Another area of study of scientific papers is the classification of their sentences. Teufel (2002) identified the rhetorical status of sentences using Bayes classifiers.

To find citation inside paper, we need to analyse the references section. Dominique Besagni et al. (2003) developed a method using pattern recognition to extract fields from the references while Brett Powley and Robert Dale (2007) looked citations and references simultaneously using informations from one task to help complete the second task.

3 Building an XML Version of the Topics

Our first concern was to make sure that once we feed the texts into our system, we could reproduce the offsets found in the training data. As there were significant differences in the format between files because of the seemingly random appearance of byte-order marks, different types of line terminators, we had to develop scripts to make all of them uniform. When we worked on the ESWC-14 task to extract information of biological paper, we found useful to have the information in XML format which allow us to extract the information inside a program using existing XML tools. To simplify subsequent access to the data, we used a set of Python scripts to transform the 11 papers of each topic into a single XML file in which the papers were identified with their roles (Research Paper or Citation Paper).

3.1 Encoding Uniformisation

We wanted to make sure that all documents were following some basic format : all files to be utf-8, without BOM and using cr-lf for the EOL. We made a checker to see if the given offset from the annotation were the one we obtain from reading the files. This helped us to find files that need modification. The BOM and ASCII problems were solved by saving the file in another format. The EOL were rectified with a `sed` command.

3.2 Sections Identification

We transformed each topic into a simple XML file containing eleven documents : The research paper (RP) and ten citation papers (CP). Each paper was divided into two sections : body and references. Inside the body, we identified the position of the abstract. The body is further divided into paragraphs and the paragraphs into sentences. Inside a sentences we identified the citations and the tokens.

An important element of this transformation was to detect the references part of the paper. The words 'References', 'Selected reading' or 'Further reading' were usually present, indicating the beginning of the references section. For 6 papers, we had to manually insert the word 'References'. The lines before the references have been identified as the 'body' of the paper.

Each line was considered a paragraph. We used the Punkt sentences tokenizer from NLTK to separate sentences and words within them. We kept the offset for each paragraph, sentences and tokens The title of the paper is the first non-empty line that doesn't start with one of these words : Article, BMC, Cell, Cover, doi, Mol, PLoS, PMCID, Published, RESEARCH, Volume, Cancer Cell.

We needed to identify the abstract. Most of the time, it is preceded by 'abstract' or 'summary'. If these words aren't present then it follows 'Open Archive'. Sometimes we found the sentence 'Get rights and content' between 'Open Archive' and the abstract.

Inside the reference section we eliminate all lines beginning with 'View Record', 'View Article', 'Full Text', 'Article |' and 'Corresponding author'.

3.3 Reference Processing

The next step separates entities in the reference section. When a reference uses more than one line, it is separated by an empty line from the next one. If not, each line corresponds to one reference.

Once references are separated, we needed to extract their marker. Some used a number, other used the name of the authors with the year of publication. For each reference, we built two kinds of markers.

The marker is the concatenation of the name, a comma and the year, adding 'and' between the names when there are two of them or adding 'et al' at the end when there are more then two. Some author's name are composed of more then one word (for example : de Cristofaro, van Leuken), in these cases two different markers were created, one with the last name and one with the full name.

It is easier to extract the information for reference that used many lines. The first line is always a marker, but it isn't always the marker that is used in the text. The next line gives the authors, the title is found on the following line. The last line had the rest of the information (year, journal, ...). There were some variations to this scheme, for instance, when the reference uses only three lines there is no title.

3.4 Citation Identification

Citation identification was a bit tricky because of the many different conventions used for references and the citations (numbers in different brackets, authors' names, etc.). To find a citation, we first needed to determine if they use a numerical form or an author/year form, are they between parentheses, brackets or nothing? We started by with the hypothesis that the marker was in name/year form. With that in mind we searched for a marker in the paper, if that was not found, we searched for the numerical form. Each search was done with a regular expression that contained the marker in parentheses and in brackets. If it is a name/year marker then it is searched without parentheses or brackets.

After having performed all this clerical work for this year's task, we realised that the XML transformation was an overkill because citations had been already identified in the input data. Still, we believe that for any future work it is better to have access to formatted text and that the time invested in this part of our software will eventually be useful for other tasks.

4 Finding Facet and Extracting References

4.1 Facet Identification

In the first task, we identify the facet of a citance and the text it points to in the RP. by finding all sentences of the RP with the same facet. We hypothesized that the reference text should be within sentences sharing the same facet as the citance. We now present how we determine the facet of a citance, then the facet for sentences in the RP and finally the reference text.

We determine the word distribution for each facet using an histogram This computation yielded a sum of each words present in each facet. To find the facet of a citance we simply needed to add the score of words that were present for each facet. The facet with the highest score was chosen for that citance.

4.2 Finding the Sentences Refered to by Citances

For that task, we want to tag a facet to sentences in the RP. In the training data, annotators had identified a facet for citation and corresponding sentences. We assume that the same facet can be attributed to the sentences they choose for reference. Using this data we train the same system as before to identify the facet of the sentences inside the RP.

To find a better list of words to ignore, we used a genetic algorithm that uses a population of list of words to take out. Each new generation will build new list from the best one in the last generation. New lists were built by removing and adding a word to an already existing list. Other lists were built by merging two sublists of the last generation.

Once the facet of the sentences were known, we needed to choose a set of sentences that represented the citance. We compared the words of the sentences with same facet as the citance with a set of words that can mesure similarity with the citance. To find these words we built 15 sets over two features : words from the abstract and/or from the citances, words in the LST and/or words not in the LST. The 3 sentences with the highest number of words in common were chosen.

4.3 Summarization

For Task2, because of time constraints, we used a simple-minded approach: we merely extracted sentences with the greatest number of words from the LST also present in the citances.

5 Results

On the training set we obtained a success rate of 47.2% for finding the facet of citance. We tried taking out each word from the computation, finding those that contributed in a positive way and those who had a negative impact on the computation (table 1).

Word not used	Success rate
human	0.43
response	0.45
development	0.45
number	0.45
expression	0.48
small	0.48
function	0.49
as	0.49

Table 1: Success rate when a word is taken out.

The set of words $P = \{$'expression', 'small', 'function', 'as' $\}$ had a positive impact when they were not used for the computation. The set of words $N = \{$'human', 'response', 'development', 'number'$\}$ had a negative impact when they were not used for the computation. We tested using different combinations of words to exclude from the computation. The result can be seen in table 2.

For finding the facet of the RP sentences, using all the words from the LST we obtained a success rate of 57.7%. Testing for each word, we were able to find 25 words that had a positive contribution when they were used and 10 words that had a negative contribution. When all the negative words are taken out we obtained a success rate of 64.8% and when only the positive ones are used we obtained 67.2%. Over 15 generation of 1000 lists each, the genetic algorithm was able to reach a success rate of 74.4%.

For the reference extraction task we considered comparing against words from the abstract and from the citance. In each case we used words belonging to the LST or/and not into the LST. The F1 score for each case are presented in table 3. The best result comes from using words not from the LST that appear in the abstract. We did a second test where we multiplied the score of a sentence by a constant if it contained the word 'we', these result are shown in table 4.

Words not used	Success rate
A	0.47
P	0.44
N	0.52
$A - P$	0.53
$A - N$	0.43
$U - P$	0.59
$U - N$	0.55

Table 2: Success rate when many words are taken out, were A is the set of words encountered in each facet, U is the set of words encountered in at least one facet, $P = \{$'expression', 'small', 'function', 'as' $\}$, and $N = \{$'human', 'response', 'development', 'number'$\}$.

In this case using words that are common with the abstract, from the LST or not, yield the best result with a constant of 1.2.

6 Discussion

We used words from the LST to identify facet of sentences in a research paper. We had received 313 citations, each annotated by four annotators. The distribution of each facet is shown in table 5 and the number of times annotators all agreed on the same facet for one citation is shown in table 5. The facet **Result** and **Discussion** were the most often used and **Hypothesis** was almost never used.

Facet	Count	Distribution
Results	490	39%
Discussion	446	36%
Method	155	12%
Implication	140	11%
Hypothesis	21	2%

Table 5: Facet distribution over 313 citances and 4 annotators

So in the best case, choosing the facet that appear the most often, we could only obtain a score of 0.66 over the training data. Our script yielded a score of 0.50, agreeing in average with two annotators out

		Abstract			
		None	Not in LST	In LST	All
Citance	None	—	0.040	0.032	0.037
	Not in LST	0.038	0.031	0.032	0.032
	In LST	0.036	0.031	0.037	0.034
	All	0.037	0.033	0.033	0.034

Table 3: F1 for different cases.

		Abstract							
		None		Not in LST		In LST		All	
		k	F1	k	F1	k	F1	k	F1
Citance	None	—	—	1.6	0.044	1.6	0.037	1.2	0.049
	Not in LST	4.0	0.045	1.4	0.041	2.6	0.045	1.4	0.042
	In LST	4.0	0.047	1.7	0.043	1.4	0.045	1.1	0.043
	All	4.1	0.045	1.4	0.042	2.4	0.046	1.4	0.043

Table 4: F1 for different cases, the first column is the constant that yield the best result and the second is the F1 score.

Facet	**4-0**	**3-1**	**2-1-1**
Results	17	53	53
Discussion	12	55	33
Method	15	12	8
Implication	1	3	3
Hypothesis	0	0	0
Total	45	123	97

Table 6: Agreement of 4 annotators over 313 citances

of four. Using only words from the LST gave good result on the biology papers, it remains to check if the same performance can be achieved on other domains.

Over the reference sentences, the same technique obtained better results, probably because of the increased number of sentences to train with. Applying this over the research paper, divided the sentences in five subsets, leaving a smaller group of sentences to look into for the reference sentences. Even with those smaller set, the task of finding the reference sentences proved to be difficult. The distinction between word from the LST didn't help either to a better score. The only positive element was the use of the word 'we' that raised the score by 10% to 30%.

7 Conclusion

We discussed how the preprocessing of the data was done to facilitate the analysis of the papers. We presented the use of distinguishing between topic and non-topic (LST) words for determining the facet of sentences in a paper. We obtained good results with a simple histogram. We described the technique for extracting the references sentences. The distinction between topic and non-topic words did not improve the results for this extraction.

References

Dominique Besagni, Abdel Belaïd, and Nelly Benet. 2003. 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

Patrick Drouin. 2010. Extracting a bilingual transdisciplinary scientific lexicon. *Proceedings of eLexicography in the 21st century : New challenges, new applications*, 7:43–54. Presses universitaires de Louvain, Louvain–a-Neuve.

Patrick Drouin. 2010. From a bilingual transdisciplinary scientific lexicon to bilingual transdisciplinary scientific colloations. *Proceedings of the 14th EURALEX International Congress*, 296–305. Fryske Akademy, Leeuwarden/Ljouwert, Pays-Bas.

Harold P. Edmundson. 1969. New Methods in Automatic Extracting. *Journal of the ACM (JACM)*, 16(2):264–285.

Aaron Elkiss, Siwei Shen, Anthony Fader, Günes Erkan, David J. States, and Dragomir R. Radev. 2008. 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

C. Lee Giles and Kurt D. Bollacker and Steve Lawrence. 1998. CiteSeer: an automatic citation indexing system. *CiteSeer: an automatic citation indexing system*, 89–98.

Kokil Jaidka, Christopher S.G. Khoo, Jin-Cheon Na, and Wee Kim Wee. 2013. Deconstructing Human Literature Reviews – A Framework for Multi-Document Summarization. *Proceedings of the 14th European Workshop on Natural Language Generation*, 125–135.

Julian Kupiec, Jan O. Pedersen, and Francine Chen. 1995. A trainable document summarizer *Research and Development in Information Retrieval - SIGIR*, 68–73.

Hans P. Luhn. 1958. The Automatic Creation of Literature Abstracts. *Ibm Journal of Research and Development - IBMRD*, 2(2):159–165.

Qiaozhu Mei, and ChengXiang Zhai. 2008. Generating Impact-Based Summaries for Scientific Literature. *Meeting of the Association for Computational Linguistics - ACL*, 816–824.

Saif Mohammad, Bonnie J. Dorr, Melissa Egan, Ahmed Hassan, Pradeep Muthukrishnan, Vahed Qazvinian, Dragomir R. Radev, and David M. Zajic. 2009. Using Citations to Generate surveys of Scientific Paradigms. *North American Chapter of the Association for Computational Linguistics - NAACL*, 584–592.

Brett Powley, and Robert Dale. 2007. Evidence-based information extraction for high accuracy citation and author name identification. *RIAO '07 Large Scale Semantic Access to Content*, 618–632.

Vahed Qazvinian, Dragomir R. Radev, Saif Mohammad, Bonnie J. Dorr, David M. Zajic, M. Whidby, and T. Moon. 2013. Generating Extractive Summaries of Scientific Paradigms. *Journal of Artificial Intelligence Research*, 46:165–201.

Advaith Siddharthan, and Simone Teufel. 2007. 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.

Simone Teufel, and Marc Moens. 2002. Summarizing Scientific Articles: Experiments with Relevance and Rhetorical Status. *Computational Linguistics - COLI*, 28(4):409–445.

Peter N. Yianilos, and Kirk G. Kanzelberger. 1997. The LikeIt Intelligent String Comparison Facility. *NEC Research Institute*.