Lexical and Syntactic cues to identify Reference Scope of Citance

Peeyush Aggarwal[1], Richa Sharma[2]

[1]Bharti Vidyapeeth College of Engineering, Delhi, India
peeyushaggarwal94@gmail.com
[2]BML Munjal University, Gurgaon, India
richa.sharma@bml.edu.in

Abstract.

In this paper, we present our system addressing Task 1 of CL-SciSumm Shared Task at BIRNDL 2016. Our system makes use of lexical and syntactic dependency cues, and applies rule-based approach to extract text spans in the Reference Paper that accurately reflect the citances. Further, we make use of lexical cues to identify discourse facets of the paper to which cited text belongs. The lexical and syntactic cues are obtained on pre-processed text of the citances, and the reference paper. We report our results obtained for development set using our system for identifying reference scope of citances in this paper.

Keywords: Natural Language Processing, Syntactic Analysis, Scientific Document Summarisation, Bag of Words

1 Introduction

The scientific research community needs different viewpoints of research contributions in summarized form. Abstract of the research contribution presents summary from the author(s) perspective. Citations of a reference paper reflect the viewpoint of the citing authors for that reference paper, and possibly in a certain context only. Summary drawn for a reference paper from its citations can put forward a different and interesting context of that reference paper. There have been several efforts towards extracting reference scope of citances, and such citations-based summary in recent years like [1], [2] etc. Kokil et al. [3] have shown through their Computational Linguistics Summarization (CL-Summ) Pilot task that citation based summary of scientific documentation is important to create for understanding different perspectives of a reference paper. Further to that pilot task, Computational Linguistics Scientific Document Summarization (CL-SciSumm-2016[1]) shared task has been designed with the goal of exploring automated summarization of scientific contributions for the computational linguistics research domain.

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

The organizers of CL-SciSumm shared task have divided the task into two parts: (1) For each citance, identify the spans of text (cited text spans) in the Reference Paper (RP) that most accurately reflect the citance, and identify the facet of the paper it belongs to; (2) Generate a structured summary of the RP from the cited text spans of the RP. Task-2 is optional. However, task-1 is required to create citations-based summary of the RP. This makes task-1 crucial and important step in creating citations-based summary of any scientific document [4]. The corpus of CL-SciSumm shared task has been created by sampling documents from ACL Anthology corpus and selecting their citing papers [9].

We have worked on task-1 ('a' and 'b') to develop our system for identifying the reference scope of the citance in the RP. We present the details of our system in Section – 2 below. This is followed by evaluation of our system, as presented in section 3, and observations in section 4. We finally present concluding remarks in Section 5.

2 Our System

In order to develop our system for the CL-SciSumm shared task, we first reviewed one sample topic (one RP and its citing papers) from the training set, and one from the development set. Manual review of these two samples revealed that though the shared task requires analysing the statements in the corpus semantically, but semantic analysis is challenging owing to the nature of the corpus. The corpus is a collection of scientific, technical articles making use of appropriate technical language, and therefore usage of varying, similar-meaning words is quite less. This makes the scope of using text semantic similarity measures quite minimal. Secondly, the citance from the citing paper refers to the text spans of RP in different contexts. These citing texts often do not refer to any meaningful content or information from RP except for a word or two. For example, the citing statement below does not convey much information about the RP except for two hinting words – *RFTagger* and *German*:

For German, we show results for ***RFTagger (Schmid and Laws, 2008)***.

Having found most of such examples in manual review, we were discouraged to make use of sub-sequences (of words) overlap between the statement in Citing Paper (CP) and its corresponding, reflective statements in RP. The overlapping words between the statements in CP and in RP usually do not form a subsequence. Therefore, we resolved to work with lexical (n-grams in bag-of-words approach instead of subsequence of words), and syntactic cues to develop our system. Following sub-sections summarize our approach and the heuristics used in our system. We have implemented our solution approach using Python. During the course of development of our system, we observed various advantages that Python offered us. We shall discuss those in observations section 3. Figure 1 below summarizes an overview of our approach implemented to develop Python-based system for carrying out task-1 of CL-SciSumm:

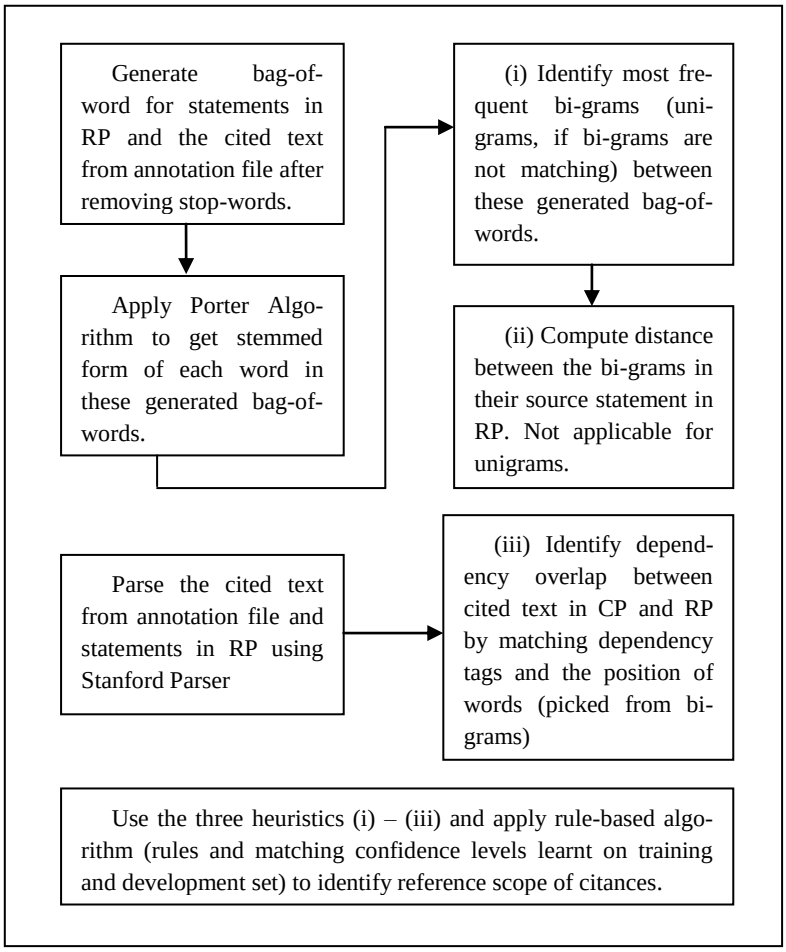

Fig. 1. System Overview

2.1 Generating Bag-of-words

Lexical cues, in our system, are gathered in terms of bi-grams (two lexical tokens from the bag-of-words) and unigrams (where bi-grams are not available). We are extending the notion of bi-grams, in our context of study, to group of two matching words between the cited text in CP and its reference scope in the RP. As discussed above, most of the citances refer to two (or more) lexical units in the reference scope of the RP. Therefore, we have limited the scope of our solution to bi-grams. We first parse the XML version of the reference paper to get individual statements in the RP for further processing. Then, we generate bag-of-words after removing stop-words from the citing text, and the statements of the RP. We have used most commonly used

Glasgow list of stop-words[2] for the purpose. We identify the (matching) bi-grams after converting the lexical units in bag-of-words to their stemmed form using Porter's Stemmer[3]. Porter's stemmer is often criticized for not returning the correct root form of a word. However, this limitation of Porter's stemmer does not affect the results in our case since we are applying it to both the bag-of-words to be used for matching (words/lexical units) purpose. Therefore, carrying out a regular expression comparison on both the bag-of-words did not add any discrepancy inadvertently.

2.2 Syntactic Dependency Analysis

Syntactic Dependency analysis has been extensively used for analysing statement at granular level for recognizing textual entailments [5] and question-answering systems [6]. Similarities in syntactic roles, and dependency overlaps between statements under analysis for semantic similarity have proved to be effective heuristics. Finding reference scope for citations could also benefit from dependency overlap, though similarity in syntactic roles is difficult to find between citations and their corresponding reference scope in RP. We have used Stanford dependency parser [7] to find dependency overlaps for the identified bi-grams between citing statement in CP and its reflective statement in RP. After obtaining parsed output, the words in the dependency relation are again converted to their stemmed form (using Porter's stemmer) to facilitate matching between different forms of same word like *use*, *using* etc.

2.3 Heuristics to identify Reference Scope of Citance

We have worked with following heuristics for task 1a of the CL-SciSumm task:

1. **Most frequent bi-grams.** We search for matching words (stemmed form) between the citing statement and the statements in RP. Having obtained a list of matching statements, we search for most frequent words in thus obtained list of statements from the RP. The count of matching words in the statements of the RP varies from zero to five-six. We observe that considering bi-grams for ranking statements in RP is if help. If none of the statements in the RP has been found to have matching words with bag-of-words of the citing statement, then we output such a citance relationship with '*NaN*'. There are instances where citing statements got stopped inadvertently due to incorrect marking of end of statement while preparing the corpus. In case only uni-gram is found to be matching between citing statement and the source statement in RP, then its weight in the matching statement is computed as the ratio of its occurrence count against the size of bag-of-words for that matching statement. The statement with highest weight is assigned rank – 1. If more than two statements have exactly same weight with same words, then all

[2] http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words
[3] https://pypi.python.org/pypi/stemming/1.0

these statements are assigned rank – 1. In case, the word are different in such similar weighing statements, then the statement having most frequent word across various statements is assigned highest rank.

In case, two or more than two words are matching, then we rank the statements considering bi-grams for weight-assignment. We identify the most frequent bi-gram across various statements that have matching words with the citing statement. For the most frequent bi-gram, we assign weight to the statement as the ratio of count of occurrence of the words in bi-gram against the size of bag-of-words in each of the source statements from RP. The matching statement having highest weight is ranked highest, and is reported as the reflective statement for the cited statement under consideration. In case, more than one statement has similar weights then ranking algorithm considers rest of the two heuristics as discussed below.

2. **Distance between tokens in frequent bi-grams.** This heuristic considers the distance between the words or tokens in the most frequent bi-gram. This heuristic helps in resolving ranks of the matching statements from the RP when the heuristic in point 1 happens to assign similar weights to more than one statement. However, this heuristic is not applicable where unigrams have been found to be matching.

3. **Dependency Overlap Count.** We determine dependency overlap for the frequent bi-grams between the cited text in CP and its corresponding reflective statement in the RP. We extract those dependencies that have either of the words or tokens in the bi-gram. A match is said to be found if the dependency tag matches, the stemmed form of tokens also match, and the token is in the identical position (either governing position or dependent position). Following example illustrates how dependency overlap is found:

   Considering the following citing statement from development set for topic, C02-1025:

   S1: In such cases, neither global features (Chieu and Ng, 2002) nor aggregated contexts (Chieu and Ng, 2003) can help.

   and one of the statements from RP:

   S2: Such a classification can be seen as a not-always-correct summary of global features.
   The parsed output for S1 and S2 is respectively:

   Parsed Output for S1 in CP:

amod(cases-3, such-2)	prep_in(help-12, cases-3)
preconj(features-7, neither-5)	amod(features-7, global-6)
nsubj(help-12, features-7)	amod(contexts-10, aggregated-9)
conj_nor(features-7, contexts-10)	nsubj(help-12, contexts-10)
aux(help-12, can-11)	root(ROOT-0, help-12)

Parsed Output for S2 in RP:

predet(classification-3, Such-1)	det(classification-3, a-2)
nsubjpass(seen-6, classification-3)	aux(seen-6, can-4)
auxpass(seen-6, be-5)	root(ROOT-0, seen-6)
amod(summay-10, not-always-correct-9)	
det(summay-10, a-8)	prep_as(seen-6, summay-10)
amod(features-13, global-12)	prep_of(summay-10, features-13)

The most frequent bi-gram for this topic is: *global, features*. Searching for these words in the parsed output of S1 and S2, we get:

S1:

preconj(features-7, neither-5)
amod(features-7, global-6)
nsubj(help-12, features-7)
conj_nor(features-7, contexts-10)

S2:

amod(features-13, global-12)
prep_of(summay-10, features-13)

Matching the dependency tags and governing/dependent positions in the above-extracted dependencies of S1 and S2, we get dependency overlap count as *one* (matching dependency presented in italics above).

2.4 Rule-based System to identify Reference Scope in RP

We have developed rule-based system based on these three heuristics to report reference scope in RP for the cited text in CP. We have learnt threshold values for ranking and further processing the statements after several rounds of experimentation with these datasets. The statements having weight more than or equal to 0.2 are considered for further ranking and processing. We report the reflective source statement in RP for a citing statement in CP after considering highest weights, maximum dependency overlap count, and lowest distance between bi-grams. In case of more than one statement encountered having similar values for any of these three heuristics, we assign weights the highest priority, followed by dependency overlap count, and then distance. After these checks, if there is still more than one statement with same values of heuristics, then our system reports all of these statements as reference scope for the CP statement.

2.5 Heuristics to identify Facets

The next sub-task of task-1 is to identify discourse facet for the cited text span with reference to the RP. The discourse facet is helpful in identifying different contexts of citing a reference paper. The organizers of the CL-SciSumm have predefined five facets: aim_citation, hypothesis_citation, method_citation, implication_citation, and

results_citation. Identifying correct discourse facet again calls for understanding semantics of the cited text span, though there are challenges involved with the same as discussed above. Our approach to identify discourse facet, therefore, is based on section headers in the paper. However, our approach has the drawback of not being able to identify hypothesis_citation and implication_citation. For rest of the three facets, following rules are observed:

1. If the cited text span lies in the introduction section, beginning of abstract, then it is indicative of aim_citation.
2. Discourse facet is marked as results_citation if the cited text span belongs to the sections having title as – Results, Observations, Discussion, Conclusion, or if the cited text span is one of the last 2 statements of the abstract.
3. If cited text span does not belong to the sections as mentioned in above two points, then the discourse facet is marked as method_citation.

3 Evaluation

We have computed ROUGE-N [8] metric with 'N' as 2 for bi-grams to evaluate our system. Table 1 below presents the average results for identifying reference scope (task – 1a) for each topic in the development set, and an average overall performance of the system for the development set for task -1a:

Table 1. Task 1a performance in terms of ROUGE-N for development set

Paper Id	Precision	Recall	F-measure
C02-1025	0.12	0.08	0.09
C08-1098	0.07	0.03	0.04
C10-1045	0.17	0.14	0.15
D10-1083	0.08	0.08	0.08
E09-2008	0.27	0.21	0.23
N04-1038	0.25	0.21	0.22
P06-2124	0.16	0.05	0.06
W04-0213	0.12	0.04	0.06
W08-2222	0.14	0.05	0.07
W95-0104	0.16	0.13	0.13
Average	0.16	0.10	0.11

For task 1b, we have computed accuracy of reporting discourse facet of the paper as the ratio of correctly identified facets in an annotation file for a topic and the total number of citances for that topic. Table 2 presents the discourse-facet accuracy corresponding to the development set:

Table 2. Discourse Facet Accuracy for development set

Paper Id	Accuracy
C02-1025	0.74
C08-1098	0.69
C10-1045	0.42
D10-1083	0.55
E09-2008	0.63
N04-1038	0.75
P06-2124	0.44
W04-0213	0.83
W08-2222	0.78
W95-0104	0.49
Average	0.63

4 Observations

The experiments with different datasets – training, development, and test set indicate that lexical and syntactic cues are indeed of help. But, lexical and syntactic analysis has its own limitations in terms of only regular expression match, and no semantic or contextual matching. We observe that the same approach does not perform uniformly with all the datasets, and performance does differ even within one dataset. For example – our system worked better with topics E09-2008 and N04-1038 as compared to other topics in development set, as evident from Table – 1. The evaluation results presented in Table – 1 correspond to ROUGE-N metric (N as 2). We have used this metric because our system is bi-gram in nature. Nevertheless, we are implementing ROUGE-S metric as well in order to cross-validate our evaluation and system performance.

It can be inferred from the discussion above that semantic-level analysis is inevitable to yield good results. The task of identifying reference scope for citances appears similar to the task of recognizing textual entailment (RTE), but is actually quite different. This is primarily because of different nature of corpus. Nevertheless, CL-SciSumm task can benefit from the RTE challenges and solution approaches to recognizing textual entailment. While working with CL-SciSumm corpus, we encountered several problems in the corpus in terms of its formatting, characters coding as well as annotations. However, these problems are not major, and could be fixed. Resolution of these concerns may provide useful pointers to semantic-level analysis needed for tasks like CL-SciSumm.

We have worked with three heuristics of lexical and syntactic nature to identify the reference scope of the cited text in the RP. The computation of values of these heuristics has been described in detail in section – 2. We observed after experiments with

our system that computation methodology of our heuristics may further be refined. As of now, our system considers unigrams and bi-grams only. We have mitigated the challenges with lexical analysis by considering stemmed form of words to work with. We are further experimenting with different priorities for our heuristics, and tweaking our algorithms currently.

We have developed our system for CL-SciSumm task in Python language. We have observed that Python turned out to be a useful choice. Python is an interpreted language supporting both object-oriented and functional programming flavour. Python allowed us to develop codes in fewer lines with dividing the problems into sub-problems. We were thus able to code and test small snippets separately and merge those later to develop complete system.

5 Conclusion

In this paper, we have presented our system for CL-SciSumm task 1 to identify reflective statements from RP for a given citance in CP. The task is challenging as semantic-level analysis has limited applicability in this case. We have addressed the task using lexical and syntactic cues to extract text-spans from RP that correspond to the cited text in CP. We believe that further refinements to the corpus and to our system can yield better results. We do intend to further refine our heuristics and check the applicability of machine learning too.

6 References

1. Nakov, P.I., Schwartz, A.S. and Hearst. M.A.: Citances: Citation sentences for semantic analysis of bioscience text. In: SIGIR (2004).
2. Qazvinian, V. and Radev, D.R.: Identifying Non-explicit Citing Sentences for Citation-based Summarization. In Proceedings of Association for Computational Linguistics, (2010).
3. Jaidka, K., Chandrasekaran, M.K., Elizalde, B.F., Jha, R., Jones, C., Kan, M., Khanna, A., Molla-Aliod, D., Radev, D.R., Ronzano, F., et al.: The computational linguistics summarization pilot task. In: Proceedings of Text Ananlysis Conference, Gaithersburg, USA, (2014).
4. Abu-Jbara, A. and Radev, D.: Reference Scope Identification in Citing Sentences. In: Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 80–90, (2012).
5. Sharma, N., Sharma, R. and Biswas K.K..: Recognizing Textual Entailment using Dependence Analysis and Machine Learning. In: Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Student Research Workshop (SRW), Colorado, USA (2015).
6. Molla, D.: Towards semantic-overlap based measures for question answering. In: Proceedings of the Australasian Language Technology Workshop, Australia (2003).

7. Marneffe, M.C. de, Silveira, N., Dozat, T., Haverinen, K., Ginter, F., Nivre, J. and Manning, C.D.: Universal Stanford Dependencies: A cross-linguistic typology. In: LREC (2014).
8. Lin, C. and Hovy, E.H.: Automatic Evaluation of Summaries using N-gram co-occurence Statistics. In: Proceedings of Language Technology Conference (HLT-NAACL), Canada (2003).
9. Jaidka, K., Chandrasekran, M.K., Rustagi, S. and Kan, M.: 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), Newark, New Jersey, USA (2016).