Recognizing reference spans and classifying their discourse facets

Kun Lu

School of Library and Information Studies
University of Oklahoma
kunlu@ou.edu

Jin Mao

School of Information
University of Arizona
danveno@163.com

Gang Li

School of Information Management
Wuhan University
imiswhu@aliyun.com

Jian Xu

School of Information Management
Wuhan University
xukeywhu@163.com

Abstract: In this shared task, we applied "Learning to Rank" algorithm with multiple features, including lexical features, topic features, knowledge-based features and sentence importance, to Task 1A by regarding reference span finding as an information retrieval problem. Task 1B, discourse facet identifying, is treated as a text classification problem by considering features of both citation contexts and cited spans.

Keywords: Learning to rank; Topic model; Facet classification

1 Introduction

The 2nd CL-SciSumm Shared task follows the TAC 2014 Biomedical Summarization Track on scientific paper summarization. An overview of the shared task, including specific details on the dataset, the competitive results and subsequent analyses for each task can be found in the shared task overview paper[1]. In this report, we provide a detailed description of the methods we used for the Task 1A and Task 1B. Our methods are introduced in Section 2 followed by results in Section 3. Some conclusions are presented in the last section.

2 Methods

2.1 Task 1A

We considered Task 1A as an information retrieval problem. A citance (citation context) is regarded as a query and sentences from reference paper (cited spans) are treated as candidate documents. Then, the problem becomes how to rank the sentences of the reference paper (i.e., candidate cited spans in this report) for a given citance. The most relevant sentences of the reference paper to a citation context are selected as the golden

sentences of the citation context. We apply "learning to rank" algorithms to address this problem and exploit multiple features. The explored features are as follows:

- **Lexical Features.** Bag of words is a widely used text representation method. By representing citation contexts and candidate cited spans with the bag of words model, the lexical similarities between citation contexts and candidate cited spans can be obtained. We chose four candidate lexical similarity features including Cosine similarity, Jaccard similarity, Dice similarity and LCS (longest common subsequence). When computing cosine similarity, TFIDF term weighing was applied. In the formula below, TF is the number of times a term occurring in a given sentence, while the IDF value of a term is computed from all the 437 papers in the training set (93 papers), development set (155 papers), and test set(229 papers). Thus, $Document\_All\_Number$ is 437, which is the same for all terms. And $Document\_number(Term)$ is the number of the different papers that a term occurs.

$$\mathrm{TFIDF(Term)} = \mathrm{TF(Term)} * \ln \frac{Document_{All_{Number}}}{Document_{number(Term)}} \qquad (1)$$

- **Topic Features.** Bag of words model is insufficient in handling polysemy and synonym problems. Taking topics into account can relieve this problem. Topic modeling method[2] was used to identify latent topics from 10,921 articles of the ACL Anthology Reference Corpus. The topic distributions of both citation contexts and candidate cited spans were predicted through the LDA models. Cosine similarity was then used to measure their topic similarities.
- **Knowledge Based Features.** WordNet[3] was used to compute the concept similarity between citation contexts and candidate cited spans. Lin similarity[4] that measures the similarity of two words was applied. The similarity of two sentences can be compute by cumulating the similarities between their words. We used different combinations of nouns and verbs to calculate similarities: WordNet (N)-only using nouns, WordNet (V)-only using verbs, WordNet (N,V)-using both nouns and verbs, and WordNet (N,Vsep) which is obtained by (WordNet(N)+ WordNet(V))/2.
- **Sentence Importance.** The importance of candidate cited spans in the reference paper is considered as a factor influencing whether they are being cited or not. TextRank[5], which is a widely used unsupervised method to extract the keyword or rank the sentences of a given document, was applied to measure the importance of a sentence in the reference paper. The assumption is that the more important a sentence is in the reference paper, the more likely the sentence belongs to the cited span.

Each pair of a citation context and a candidate sentence from the cited article is an instance. Positive instances (i.e., the candidate cited sentence is one of the gold standard cited sentences) were assigned higher scores than negative instances. The above features were calculated and fed into "learning to rank" methods. For topic similarity, the number of topics varied from 20 to 200 with a step of 20. The features showing high performance were selected as the final set of features. Then, five learning to rank algorithms from RankLib[6], including RankBoost[7], RankNet[8], AdaRank[9], and Coordinate Ascent[10], were compared.

2.2 Task 1B

Task 1B is considered as a text classification problem. The five discourse facets of a sentence in a reference paper are Aim, Method, Result, Implication, and Hypothesis. The features adopted by the classifiers are as follows.

- The text of the reference sentence
- The title of the section that the reference sentence belongs to
- The section type of the section that the reference sentence belongs to
- The text of the citation context
- The title of the section that the citation context belongs to
- The section type of the section that the citation context belongs to

Three classifiers from Weka[11] including Naïve Bayes, Decision Tree and Supporting Vector Machine were applied and compared for Task 1B. The algorithm behind these classifiers are Naive Bayesian classification, C4.5, and Sequential Minimal Optimization. Default parameter settings were used to train the classifiers implemented in Weka.

2.3 Evaluation

2.3.1 Task 1A metrics.

Two groups of precision (P), recall (R) and F_1 measures were used to evaluate the performance in Task 1A. The first group counted the number of sentences returned by our methods that match the gold standard sentences annotated by the task organizers.

$$R = \frac{|G \cap S|}{|G|} \quad P = \frac{|G \cap S|}{|S|} \quad \mathrm{F}_1 = \frac{2 * R * P}{(R + P)} \qquad (2)$$

where G indicates the gold standard sentences, S denotes the sentences returned by our methods.
The second group of measures are ROUGE_1[12] (Recall-Oriented Understudy for Gisting Evaluation) measures used as the official comparison measures.

2.3.2 Task 1B metrics.

Precision (P), recall (R) and F_1 measures were calculated to evaluate the performance for each facet. For one facet, *a* is denoted as the number of correct predictions for this facet, *b* is the number of wrong predictions for this facet, and *c* is the number of cited sentences from gold standard sentences for this facet that are not predicted as belonging to this facet.

$$\mathrm{R} = \frac{\mathrm{a}}{\mathrm{a+c}} \quad \mathrm{P} = \frac{\mathrm{a}}{\mathrm{a+b}} \quad F1 = \frac{2*\mathrm{R}*\mathrm{P}}{(\mathrm{R+P})} \qquad (3)$$

Then, Macro average and Micro average were used to evaluate the system on all facets. Macro average measures are computed as:

$$\mathrm{R}_{\mathrm{macro}} = \frac{\sum_f^N R_f}{\mathrm{N}} \quad P_{macro} = \frac{\sum_f^N P_f}{\mathrm{N}} \quad F1_{macro} = \frac{2*\mathrm{R}_{macro}*P_{macro}}{R_{macro}+P_{macro}} \tag{4}$$

Micro average measures are computed as:

$$R_{micro} = \frac{\sum_f^N a_f}{\sum_f^N a_f+\sum_f^N c_f} \quad P_{micro} = \frac{\sum_f^N a_f}{\sum_f^N a_f+\sum_f^N b_{f_i}} \quad F1_{micro} = \frac{2*\mathrm{R}_{micro}*P_{micro}}{R_{micro}+P_{micro}} \tag{5}$$

where *f* is the facet index, *N* is the number of facets (i.e., 5). A good classifier should have both high Macro and Micro average measures.

3 Results

In this section, the results where the training set is used for training and the development set is used for testing.

3.1 Task 1A results

According to Fig.1, Jaccard similarity and Dice similarity achieved better F_1 measures than Cosine similarity and LCS. Topic similarity feature (indicated as Topic_number in Fig.1) showed slight different performance among different numbers of topics. WordNet based similarity achieved best results when combining the results of standalone nouns similarities and verb similarities. TextRank showed similar results to topic similarity and WordNet based similarity, however, this feature did not improve the performance of learning to rank. Finally, Jaccard similarity, Topic similarity with 200 topics, WordNet (N,Vsep), and TextRank were kept. In the development set, Rank-Net and AdaRank achieved best performance with Overlap/ ROUGE_1 F_1 of 0.057/0.211 (Table 1).

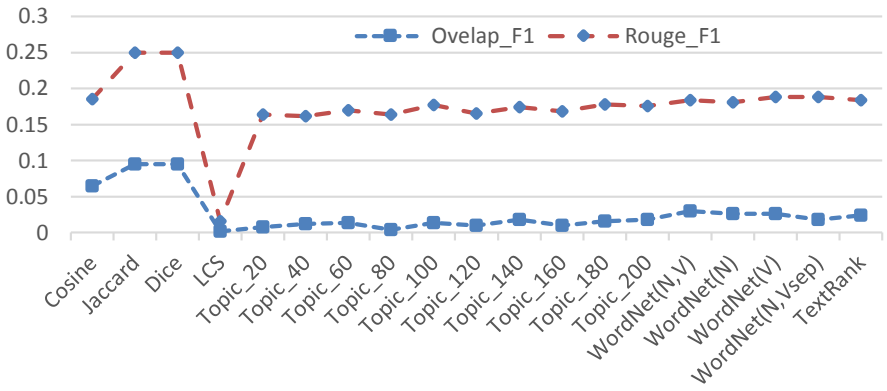

Figure 1. The F_1 values for each ranking feature on development set

Table 1. The performance of different learning to rank algorithms on development set

Learning to rank Algorithms	Overlap			ROUGE_1		
	P	**R**	**F_1**	**P**	**R**	**F_1**
RankBoost	0.008	0.015	0.010	0.129	0.171	0.147
RankNet	**0.043**	**0.085**	**0.057**	**0.194**	**0.232**	**0.211**
AdaRank	**0.043**	**0.085**	**0.057**	**0.194**	**0.232**	**0.211**
Coordinate Ascent	0.012	0.024	0.016	0.153	0.190	0.169

3.2 Task 1B results

Table 2 lists results for Task 1B. SVM showed best micro average performance (F_1=0.657), while Naïve Bayes achieved the best macro average performance (F_1=0.269). It's observed that Naïve Bayes showed balanced performance over the five facets. Both Decision Tree and SVM had poor prediction on Aim, Implication and Hypothesis facets.

Table 2. The performance of the three classifiers on development set

Facets	# of Cited Spans	Naïve Bayes			Decision Tree			SVM		
		P	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**
Aim	29	0.197	0.482	**0.280**	0.023	0.034	0.028	0	0	0
Method	142	0.683	0.578	0.626	0.625	0.634	0.629	0.671	0.950	**0.787**
Result	33	0.407	0.333	0.367	0.615	0.484	**0.542**	0.75	0.272	0.400
Implication	8	0.006	0.015	**0.008**	0	0	0	0	0	0
Hypothesis	7	0	0	0	0	0	0	0	0	0

Micro Avg	-	0.488	0.488	0.488	0.488	0.488	0.488	0.657	0.657	**0.657**
Macro Avg	-	0.259	0.282	**0.269**	0.253	0.230	0.240	0.284	0.244	0.262

4 Final run methods and conclusions

For the last run, we used both training set and development set to train instances. We chose Jaccard, Topic_200, WordNet (N+V) and TextRank as the final ranking features and applied AdaRank learning to rank algorithm on Task 1A. For Task 1B, Naïve Bayes was used as the classifier.

Recognizing the cited spans and determining their discourse facets are very challenging for the summarization of scientific papers. There are some issues to be addressed during our study on the tasks. First, either cited span discovery or discourse facet classification is an unbalanced problem—negative examples are more prevalent than positive examples. That would be our future work. Second, Task 1A involves much more than a similarity problem. This is because the underlying citation intentions are complex. More features that reflect the citation intentions should be explored.

References

[1]Kokil Jaidka, Muthu Kumar Chandrasekaran, Sajal Rustagi, and Min-Yen Kan (2016). 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.

[2]Blei, D. M. (2012). Probabilistic topic models. Communications of the ACM, 55(4), 77-84.

[3]George A. Miller (1995). WordNet: A Lexical Database for English. Communications of the ACM Vol. 38, No. 11: 39-41.

[4]Lin, D. (1998, July). An information-theoretic definition of similarity. In ICML (Vol. 98, pp. 296-304).

[5]Mihalcea, R., & Tarau, P. (2004, July). TextRank: Bringing order into texts. Association for Computational Linguistics.

[6]Dang, V. The Lemur Project-Wiki-RankLib. Lemur Project,[Online]. Available: https://sourceforge.net/p/lemur/wiki/RankLib.

[7]Freund, Y., Iyer, R., Schapire, R. E., & Singer, Y. (2003). An efficient boosting algorithm for combining preferences. The Journal of machine learning research, 4, 933-969.

[8]Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., & Hullender, G. (2005, August). Learning to rank using gradient descent. In Proceedings of the 22nd international conference on Machine learning (pp. 89-96). ACM.

[9]Xu, J., & Li, H. (2007, July). Adarank: a boosting algorithm for information retrieval. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 391-398). ACM.

[10]Metzler, D., & Croft, W. B. (2007). Linear feature-based models for information retrieval. Information Retrieval, 10(3), 257-274.

[11]Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, 11(1), 10-18.
[12]Lin, C. Y. (2004, July). Rouge: A package for automatic evaluation of summaries. In Text summarization branches out: Proceedings of the ACL-04 workshop (Vol. 8).