PolyU at CL-SciSumm 2016

Ziqiang Cao[1], Wenjie Li[1], and Dapeng Wu[2]

[1] Department of Computing, The Hong Kong Polytechnic University, Hong Kong,
{cszqcao, cswjli}@comp.polyu.edu.hk
[2] Department of Electrical & Computer Engineering, University of Florida, USA,
wu@ece.ufl.edu

Abstract. This document demonstrates our participant system PolyU on CL-SciSumm 2016. There are three tasks in CL-SciSumm 2016. In Task 1A, we apply SVM Rank to identify the spans of text in the reference paper reflecting the citance. In Task 1B, we use the decision tree to classify the facet that a citance belongs to. Finally, in Task 2, we develop an enhanced Manifold Ranking summarization model.

1 Introduction

The CL-SciSumm Shared Task [2] at BIRNDL 2016 (http://wing.comp.nus.edu.sg/birndl-jcdl2016/) focuses on automatic paper summarization in the Computational Linguistics (CL) domain. A document set of CL-SciSumm consists of a Reference Paper (RP) and Citing Papers (CPs) that all contain citations to the RP. In each CP, the text spans (i.e., citances) have been identified that pertain to a particular citation to the RP. Given this dataset, a participant system is expected to handle three tasks. **Task 1A**: For each citance, identify the spans of text (cited text spans) in the RP that most accurately reflect the citance. These are of the granularity of a sentence fragment, a full sentence, or several consecutive sentences (no more than 5). **Task 1B**: For each cited text span, identify what facet of the paper it belongs to, from a predefined set of facets. **Task 2** (optional): Finally, generate a structured summary of the RP from the cited text spans of the RP. The length of the summary should not exceed 250 words.

Our system PolyU implements all the three tasks. For Task 1A, we treat it as a ranking problem modeled by SVM Rank [3]. For Task 1B, since the facet distribution is extremely imbalanced, the Decision Tree Classifier is introduced to naturally conduct the task as hierarchical classification. For the final summarization task, we treat these CPs as queries and each section as a document. The idea behind is that CPs often refer to important sentences and sentences in important sections should also be important. Then we improve the widely-used query-focused summarization model Manifold Ranking [4] to generate summaries. Manifold Ranking can naturally make full use of both the relationships among sentences in different sections and the relationships between the CPs and the sentences. We introduce an extra parameter in Manifold Ranking to adjust the weights for CPs. The overall performance of PolyU is presented in Table 1.

Task	Performance
1A	Accuracy: 11.8, Recall: 8.7, F1-score: 10.0
1B	Micro-accuracy: 61.9, Macro-accuracy: 21.4
2	ROUGE-1: 49.5, ROUGE-2: 15.4

Table 1. Overall performance (%) of PolyU.

2 Task 1A

2.1 Problem Transformation

In this task, we need to identify the spans of text (cited text spans) in the RP that most accurately reflect the citation. Since a citation text span can be linked to many sentences in the reference paper, we firstly analyze the corresponding sentence number distribution, as shown in Fig. 1. As can be seen, a large proportion of reference spans contain more than one sentences. Therefore, the most direct approach for this task is to train a ranking model and select a series of top ranked sentences. However, this idea has two disadvantages. On the one hand, the threshold is hard to set due to the serious variation of ranking scores on different document sets. On the other hand, a reference span tends to contain adjacent sentences, which cannot be reflected by the top ranked items. The adjacency property of reference spans is presented in Fig. 2. In this figure, we count the number of reference sentences which fail to be covered by adjacent sentence chunks. We observe that most multi-sentence reference spans are just a pair of adjacent sentences. Meanwhile, when the chunk size ≥ 4, about 90% reference sentences can be covered by sentence chunks, and the uncovered ratio keeps stable.

Therefore, we simplify Task 1A into a ranking problem which just needs to select the first item. Specifically, like n-grams, we put adjacent sentences into n-sentence chunks. According to the data property, we set $n \in [1, \cdots, 4]$. Then, a reference paper is represented by these n-sentence chunks. The actual ranking score of an n-sentence chunk is the ratio of reference sentences it contains. We extract a series of features from (CP, n-sentence chunk) pairs. Finally, we train a ranking model and choose the top ranked n-sentence chunk as the reference span.

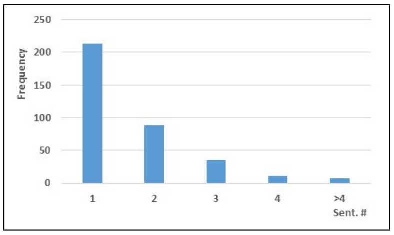

Fig. 1. Reference sentence number distribution on the training set.

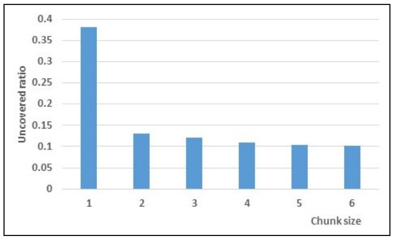

Fig. 2. Numbers of uncovered sentences with the change of chunk sizes.

2.2 Model Description

Since most sentence chunks have no reference sentences, the regression model trained on this dataset tends to predict the zero score. Therefore, we adopt SVM Rank [3] to handle this ranking task. SVM Rank is a popular supervised pair-wise model. It converts a ranking task into a binary classification task, which avoids the problem of data imbalance. The major feature we use is the tf-idf cosine similarity between a citance and a sentence chunk in the reference paper. We also extract some citance-independent features such as the position of the sentence chunk. The motivation behind is that most citances are related to the facet of method. As a result, the reference sentences may have some common characteristics. The whole ranking features are presented in Table 2.

We analyze the model weight for each feature. The feature SIMILARITY holds the highest weight, which accords with common sense. In addition, three position features, i.e., SENT_POSITION, SECTION_POSITION and INNER_POSITION all have relatively large negative weights. It seems sentences in front are more likely to be cited.

Name	Description
SIMILARITY	The tf-idf cosine similarity between a citance and a sentence chunk.
SENT_POSITION	The sentence position, divided by the number of sentences.
SECTION_POSITION	The position of the corresponding section of the sentence chunk, divided by the number of sections
INNER_POSITION	The sentence position in the section, divided by the number of sentences in the section.
NUMBER	Indicate whether there are numbers in the sentence chunk.
NER	Indicate whether there are named entities in the sentence chunk.

Table 2. Ranking features.

3 Task 1B

3.1 Problem Transformation

In this task, we need to identify what facet of the paper a citance belongs to, from a predefined set of facets. In total, there are 5 facets, i.e., Method, Aim, Results, Implication and Hypothesis. Notably, a citance can be labeled by multiple facets. Thus, it is a multi-label classification task. However, from the training data, we find more than 90% of citances only belong to one facet. Therefore, we simply treat this task as the common multi-class classification problem by reserving the first facet for citances with more than one facets.

Afterwards, we analyze the facet distribution on the training set. The result is shown in Fig 3. From this figure, we find the facet distribution is extremely imbalanced. The Method facet takes about 60% proportion of the total data, and there are only 9 instances in the Hypothesis facet. Trained on the extremely biased dataset, many classifiers such as SVM and Naive Bayes tend to classify all the data into the Method facet. Although this practice achieves high overall accuracy, it is not a proper solution. Therefore, we introduce two metrics to measure the performance, i.e., the macro-averaged accuracy (A_M) as well as the micro-averaged accuracy (A_m). Their formulas are as follows:

$$A_m = \frac{\sum_{c \in C} r_c}{\sum_{c \in C} N_c} \tag{1}$$

$$A_M = \frac{1}{|C|} \sum_{c \in C} \frac{r_c}{N_c} \tag{2}$$

where C stands for the class set, r_c is the right number in the Class c, and N_c is the actual number. Just predicting the Method facet, $A_m = 0.59$ and $A_M = 0.2$.

We focus on the increase of the macro-averaged accuracy. To this end, we use the decision tree to conduct hierarchical classification. The most important advantage of the decision tree is that it has the ability to remember patterns of all the facets in the training data. In comparison, SVM and Naive Bayes are likely to merely reserve the patterns of the dominant class.

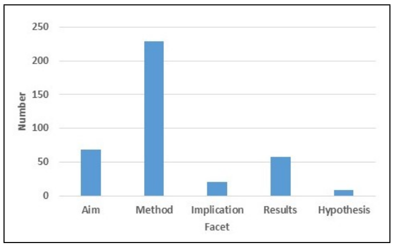

Fig. 3. The distribution of facets.

3.2 Model Description

Since the instances for the Implication and Hypothesis facets are very limited, we only train the classification model on the data of the other three facets. We use the tf-idf vector of the citance as features. Notably, the vocabulary size is too large with respect to the training data. We thereby introduce the χ^2-test to reserve the most significant 125 features. The learned decision tree is displayed in Fig. 4. We can find in the leaf nodes, there is often just one support case. This decision tree seems still quite overfitted. We also consider to add more features such as the position of the citance in the citation paper. However, the result shows these features only make the decision tree more biased to the Method facet.

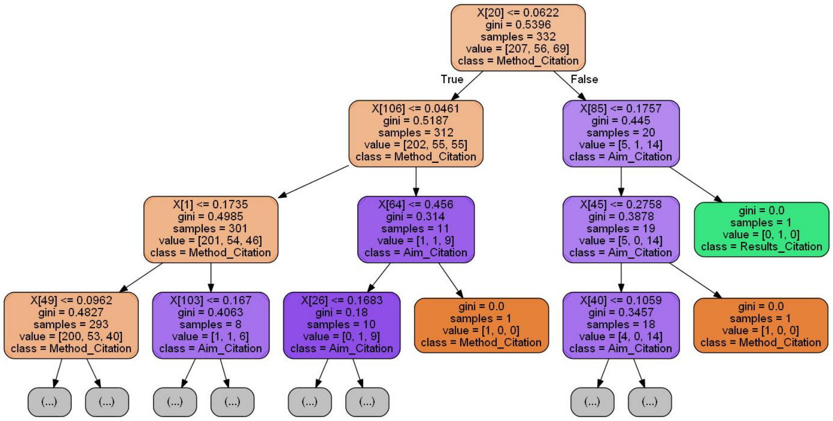

Fig. 4. Decision tree classifier. Each feature $X[.]$ stands for a tf-idf score.

4 Task 2

This task requires to generate a summary for the reference paper with the help of citances. Our summarization system makes full use of the structure information in the corpus. We regard a section of a paper as a document, since sentences in important sections should also be important. Meanwhile, we treat the citance as a query. The idea behind is that sentences relevant to citance may be the focus of the paper. After the above two steps, Task 2 is converted into the query-focused multi-document summarization problem. Then we develop an enhanced version of Manifold Ranking [4] to generate summaries.

4.1 Enhanced Manifold Ranking

Manifold Ranking [4] can naturally make full use of both the relationships among all the sentences in the documents and the relationships between the given query and the sentences. In Manifold Ranking, a document set is represented by a sentence list $D = \{x_1^q, \cdots x_k^q, x_{k+1}^D, \cdots x_n^D\}$, where x_i^q represents a query sentence and x_i^D stands for a document sentence. Then, we compute the similarity matrix $W \in R^{n \times n}$, where W_{ij} is the tf-idf cosine similarity of two sentences. Different from LexRank [1], Manifold Ranking distinguishes the relationships between inter-document and intra-document sentences. Specifically, W can be decomposed as:

$$W = W_{\text{inter}} + W_{\text{intra}} \tag{3}$$

Then, it gives different weights for these two matrices.

$$\widetilde{W} = \lambda_1 W_{\text{inter}} + \lambda_2 W_{\text{intra}} \tag{4}$$

We fix $\lambda_1 = 1$, and change λ_2 in the experiments. If $\lambda_2 < 1$, inter-document links are more important than the intra-document links in the algorithm and vice versa. Note that if $\lambda_2 = 1$, Equation 4 reduces to Equation 3.

Subsequently, We normalize the similarity matrix $\widetilde{W}$ into a probability matrix S.

$$S = G^{-1/2}\widetilde{W}G^{-1/2} \tag{5}$$

where G is the diagonal matrix with (i,i)-element equal to the sum of the i_{th} row of $\widetilde{W}$. With the probability matrix S, we can now apply the random walk algorithm to compute the saliency scores f of the sentences:

$$f(t+1) = \alpha S f(t) + (1-\alpha)y, \tag{6}$$

where α is a weight parameter, and y is the prior score distribution. In Manifold Ranking, y is set as follows:

$$y_i = \begin{cases} 1/k, & i \leq k \\ 0, & otherwise \end{cases} \tag{7}$$

It means the query-relevant sentences are most important in the prior.

We improve Manifold Ranking by modifying the prior score distribution to inspect the importance of citances. We introduce an extra parameter $\gamma \in [0,1]$ to control the weight of citances, and Eq. 7 becomes:

$$y_i = \begin{cases} \gamma/k, & i \leq k \\ \frac{1-\gamma}{n-k}, & otherwise \end{cases} \tag{8}$$

Maniflod Ranking is a special case where $\gamma = 1$.

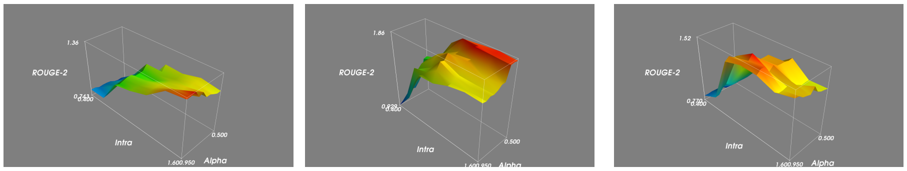

Fig. 5. Grid search with $\gamma = 0, 0.5, 1$ respectively.

4.2 Parameter Selection

There are three parameters in our summarization model, i.e., the intra-document weight λ_2, the random walk weight α and the citance weight γ. We set $\gamma = 0, 0.5, 1$ respectively and conduct grid search on the development set to check the change of performance. The result is shown in Fig 5. As can be seen, $\gamma = 0.5$ shows the highest performance potential. Meanwhile, when $\lambda_2 > 0.6$, the performance is not sensitive to its change. For α, a common value of 0.85 often works well. To sum up, we choose $\gamma = 0.5$, $\lambda_2 = 0.8$ and $\alpha = 0.85$ for the test dataset.

5 Conclusion

This document demonstrates our participant system PolyU on CL-SciSumm 2016. There are three tasks in CL-SciSumm 2016. In Task 1A, we apply SVM Rank to identify the spans of text in the reference paper reflecting the citance. In Task 1B, we use the decision tree to classify the facet that a citance belongs to. Finally, in Task 2, we develop an enhanced Manifold Ranking summarization model.

References

1. Erkan, G., Radev, D.R.: Lexrank: Graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research pp. 457–479 (2004)
2. Jaidka, K., Chandrasekaran, M.K., Rustagi, S., Kan, M.Y.: Overview of the 2nd computational linguistics scientific document summarization shared task (cl-scisumm 2016). In: Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2016) (2016)
3. Joachims, T.: Training linear svms in linear time. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 217–226. ACM (2006)
4. Wan, X., Yang, J., Xiao, J.: Manifold-ranking based topic-focused multi-document summarization. In: IJCAI. vol. 7, pp. 2903–2908 (2007)