NJUST @ CLSciSumm-18

Shutian Ma[1], Heng Zhang[1], Jin Xu[1], Chengzhi Zhang[1,2,*]

[1] Department of Information Management, Nanjing University of Science and Technology, Nanjing, China, 210094
[2] Fujian Provincial Key Laboratory of Information Processing and Intelligent Control (Minjiang University), Fuzhou, China, 350108
mashutian0608@hotmail.com, 525696532@qq.com, xujin@njust.edu.cn, zhangcz@njust.edu.cn

Abstract. This paper introduces NJUST system which is submitted in CL-SciSumm 2018 Shared Task at BIRNDL 2018 Workshop. The training corpus contains 40 articles which are created by randomly sampling documents from ACL Anthology corpus and selecting their citing papers. Overall, there are three basic tasks in CL-SciSumm 2018. Task 1A is to identify cited text spans in reference paper. Briefly, we use multi-classifiers and resemble their results via voting system. Meanwhile, we also submit results generated via single classifiers. For task 1B, which is to identify facets of cited text, except rule-based methods using human-labeled and POS dictionary, we also apply supervised topic modeling and gradient boosted decision trees. As to Task 2, after organizing identified sentences into groups based on their similarities between abstract sentences, we rank them using several features and generate a summary within 250 word by selecting the top ones.

Keywords: Cited Text Span Identification, Multi-classifiers, Voting System, Automatic Summarization, Scientific Summarization.

1 Introduction

Nowadays, increasement of publications makes researchers hard to catch up with the progress in fields. In order to provide readers a quick overview of papers, scientific summarization has arisen people's attentions. Since citation sentences (citances) usually provide useful information about reference papers, researchers were focusing on citation-based summarization by aggregating all citances that cite one unique paper [3]. However, detailed information cannot be revealed enough in citation texts, and viewpoints of the citing authors can also be different from each other due to citing purposes [4]. Recently, a number of shared tasks like, TAC 2014 Biomedical Summarization Track[1], Computational Linguistics Scientific Document Summarization Shared Task

* Corresponding Author.
[1] Available at: https://tac.nist.gov/2014/BiomedSumm/index.html

(CL-SciSumm 2016[2], CL-SciSumm 2017[3] and CL-SciSumm 2018[4]) are proposed to do summarizations based on cited text spans, which is different from traditional methods. Since the summaries are built based on reference paper itself, they are expected to provide reliable context information than citances. In this paper, we want to describe our system submitted in CL-SciSumm 2018. Basically, there are two main parts in CL-SciSumm shown in Figure 1, Task 1A is to identify cited text spans in reference paper. Task 1B is to do facet identification and summary generation is finally done in Task 2.

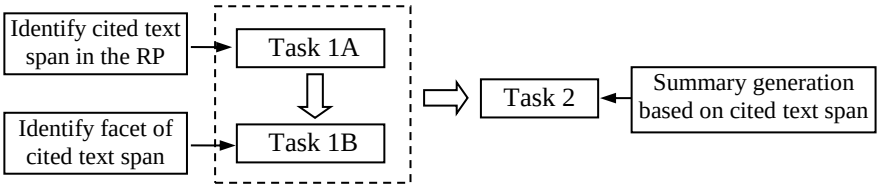


Fig. 1. Framework of CL-SciSumm Shared Task

Below is the detailed information of tasks.

Given: *A topic consisting of a Reference Paper (RP) and Citing Papers (CPs) that all contain citations to the RP. In each CP, the citances have been identified that pertain to a particular citation to the RP.*

Task 1A: *For each citance, identify the cited text span 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: *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.*

Referring to our previous work in CL-SciSumm 2017 [5], multiple classifiers are integrated based on a weighted voting system to identify cited text spans. Based on that, we did some optimizations for Task 1A from aspects of feature selection, class-imbalanced data processing, voting weights allocation and parameter tuning [6]. While in system applied in CL-SciSumm 2018, we conduct the similar strategy with multi-classifiers in Task 1A, but adding new steps to process data, new features for classifiers and new classifiers as well. For Task 1B, we try to identify facet by supervised topic modeling and classifier except using built dictionaries. Final results are combined between strategies. When doing summarization in Task 2, we firstly separate sentences based on their similarity to abstracts and rank them over several features to select important ones for summary generations.

The rest of paper is organized as follows. Section 2 provides a brief review of related works. Section 3 elaborates the detailed information about our system this year. Experimental data and evaluation results on training data are given in section 4. Conclusion and direction for future research are outlined in section 5.

---

[2] Available at: http://wing.comp.nus.edu.sg/cl-scisumm2016/

[3] Available at: http://wing.comp.nus.edu.sg/~cl-scisumm2017/

[4] Available at: http://wing.comp.nus.edu.sg/~cl-scisumm2018/

2 Related Work

With million publications are coming out every year [7], attention has been paid in automatic scientific summarization due to people's demand for getting quick overviews. Recently, Computational Linguistics Scientific Document Summarization Shared Task are the first annual medium-scale shared task on scientific summarization, where summary is generated from identified cited text. This year, CL-SciSumm 2018 took place at the Joint Workshop on BIRNDL 2018[5] with the same goal of exploring automated summarization of scientific contributions for computational linguistics domain. Here, we do literature review of different tasks based on submitted systems in CL-SciSumm 2016 and CL-SciSumm 2017 [8].

Looking at the related work of Task 1A, most teams solved it by characterizing the linkage between a citance in citing paper and its corresponding cited text spans in reference paper [9]. Features are basically generated based on character-based and semantic-based similarities. For example, in CL-SciSumm 2016, CIST System applied lexical similarity and sentence similarity [10]. Aggarwal and Sharma [11] made use of subsequences overlap. PolyU utilized TF-IDF cosine similarity, position of sentence chunk and some lexical rules [12]. Other relevant features applied in CL-SciSumm 2017 are longest common subsequence [13], character-level TF-IDF scores [14], modified Jaccard distance [15]. Deep learning methods for sematic measurement between sentences, such as pairwise neural network ranking model [13], popular word embedding models like Word2Vec and Doc2Vec [5] were also used. In order to find the most similar sentence pair, SVM and its modification model were chosen as the classifier for many teams [10, 12, 16]. Except applying one single model [17, 18], nearly half of teams applied weighted voting algorithms to integrate results [5, 13, 15].

As for Task 1B, proportions of different discourse facet types are very imbalanced, most proposed methods are using rule-based methods, which is based on human-labeled dictionaries or some heuristics. Aggarwal and Sharma [11] identified the facet based on cited text span location, such as if cited text span lies in introduction section, beginning of abstract, then it is indicative of aim citation. CIST System took advantages of frequent word and combined it with subtitle to do judgements[10, 14]. Besides, different classifiers are also applied here, such as random forest classifier[19], SVM [14], SMO [20], convolutional neural networks [17]and so on. Except position and similarity features, new ones are proposed, like Dr inventor sentence related features and scientific gazetteer features in [15].

When doing Task 2, basically, there are two main steps. First is to cluster identified text spans to organize them into groups. Second is to rank them based on different features, which depict sentence importance in some level. CIST system calculated sentence scores of five features [10]. In order to control redundancy of summary, they used determinant point processes to enhance diversity [14]. Abura'ed, Chiruzzo [15] proposed a modified version of 2016 summarization system with additional features which are relevant with reference paper and citing paper.

[5] Available at: http://wing.comp.nus.edu.sg/~birndl-sigir2018/

3 Methodology

As mentioned in introduction, there are two tasks. The dataset comprised 40 annotated sets of references and their citing papers from the open access research papers in the computational linguistics domain. A topic is consisted of a Reference Paper (RP) and Citing Papers (CPs) that all contain citations to the RP. In each CP, the text spans (citances) have been identified that pertain to a particular citation to the RP.

3.1 Task 1A

In this paper, we solve Task 1A by finding the sentence in RP that is more similar with citance. There are two main steps in our system: selecting suitable features for classifiers, integrating final results via a weighted voting system. Here are the detailed information about our system for conducting Task 1A.

Citation Text Preprocess. Since training data is labeled by human which might have some errors, we utilize two rules to expand labeled citation text in advance which can rich semantic information of citation text: First, if the next sentence behind labeled citation text contains the same author name in citation text (Example in Paper [1]), then we add this sentence into citation text. Second, if the next sentence behind labeled citation text contains demonstrative pronouns (Example in Paper [2]), then we add this sentence into citation text. We do this preprocess on training and testing data directly. For training data, there are 4,244 sentences are added into original citation texts.

Paper [1]

Like others, we have assumed lexical semantic classes of verbs as defined in ***Levin (1993)*** (hereafter Levin), which have served as a gold standard in computational linguistics research (Dorr and Jones, 1996; Kipper et al., 2000; Merlo and Stevenson, 2001; Schulte im Walde and Brew, 2002). ***Levin's*** classes form a hierarchy of verb groupings.

Paper [2]

The system described in this paper is similar to the MENE system of (Borthwick, 1999). ***It*** uses a maximum entropy framework and classifies each word given its features.

Fig. 2. Examples when Utilizing Rules to Expand Labeled Citation Text

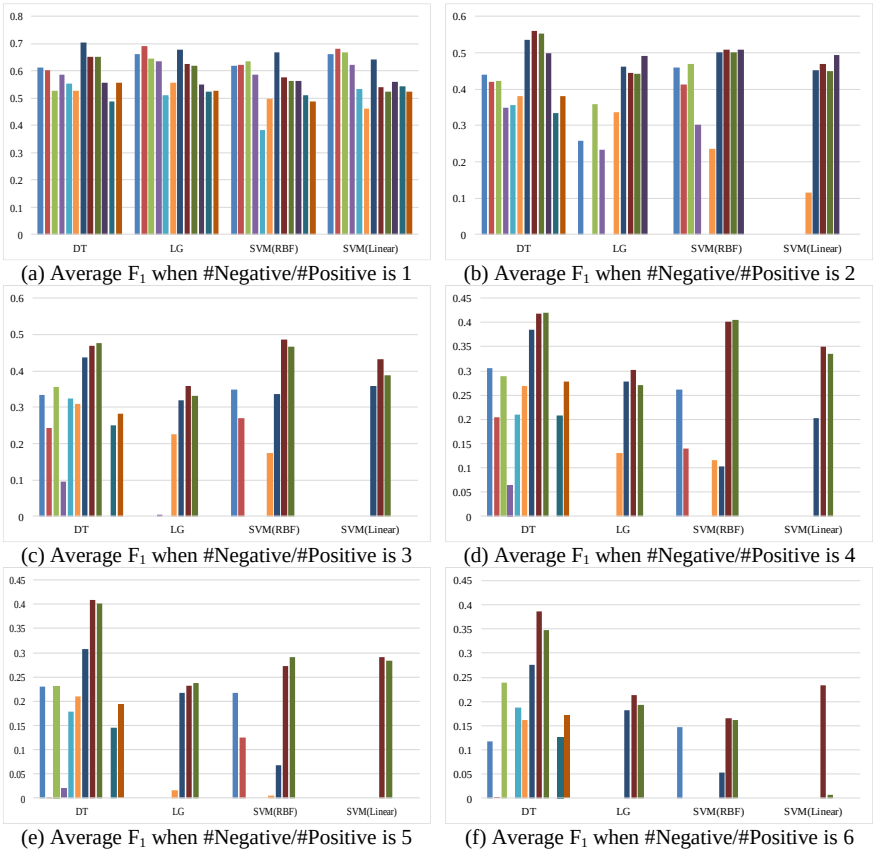
Feature Selection. Similar with previous system in CL-SciSumm 2017, we applied three kinds of features to figure out linkage between sentences in scientific papers, they are similarity-based features, rule-based features and position-based features. Then different kinds of features are generated for measuring linkages between citations and cited text. In previous work [5], bi-gram feature didn't work well, in order to convert this feature into an efficient one, we count frequency of bi-grams in training data and build a dictionary containing all the bigrams that frequency is over 500. When we find the same bigram contained in citation sentence and reference sentence, we will filter

them based on this dictionary. For sentence similarity features, we add WordNet similarity and Word2Vec similarity, which are the average of word pair similarities, whose words are contained in the two sentences. Table 1 gives the short descriptions of features we utilized in this task.

Table 1. Three Kinds of Features Applied in Four Classifiers

Feature Type	Feature	Feature Definition
Similarity-based features	LDA similarity	Cosine value between two sentence vectors trained by LDA (Topic number is set to be 20, iteration times is 2000.)
	Jaccard similarity	Division between the intersection and the union of the words in two sentences
	IDF similarity	Add up *IDF* values of the same words between two sentences
	TF-IDF similarity	Cosine value between two sentence vectors represented by *TF-IDF* (Sentence vectors haven't done normalization.)
	Doc2Vec similarity	Cosine value between two sentence vectors trained by Doc2Vec (Distributed representation vector is set to be 200)
	WordNet similarity	Average of word pair similarities calculated via WordNet
	Word2Vec similarity	Average of word pair similarities calculated via Word2Vec (Distributed representation vector is set to be 300)
Rule-based features	Filtered Bigram	After filtering, bi-gram matching value, if there is any of bi-gram matched between two sentences, this value is 1; otherwise 0.
Position-based features	Sid	Sentence position in the full text
	Ssid	Sentence position in the corresponding section
	Sentence 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

To select relevant features for use in model construction, we firstly tested each feature with four classifiers, including Decision Tree (DT), Logistic Regression (LR), SVM (kernel function is linear and RBF). We select negative and positive samples in different class ratios: 1, 2, 3, 4, 5 and 6 to investigate performance stability using different training datasets. Figure 2 displays the average F_1 values of different feature-classifier combinations.



(a) Average F_1 when #Negative/#Positive is 1
(b) Average F_1 when #Negative/#Positive is 2
(c) Average F_1 when #Negative/#Positive is 3
(d) Average F_1 when #Negative/#Positive is 4
(e) Average F_1 when #Negative/#Positive is 5
(f) Average F_1 when #Negative/#Positive is 6

Fig. 3. Average F_1 of All Features with Different Proportion of Negative and Positive Samples

In order to pick out the best feature combinations, we conduct subset selection by iteratively evaluating a candidate subset of selected features set. Based on Figure 2, for each classifiers, we choose features which are the most robust among different class ratios and have good performance to be the fixed features. Less robust features are selected to be the selected features set. We set class ratios of negative and positive samples to be 5.5. Table 2 to Table 5 shows the fixed feature and selected feature sets for each classifier and their performance of precision, recall, F_1.

Table 2. Fixed and Selected Feature Sets for SVM (Linear) and their Precision, Recall, F_1

Fixed Features	Selected Features	P	R	F_1
tfidf_sim, idf_sim		0.2231	0.0216	0.0391
	bigram	0.5356	0.0647	***0.1140***
	lda_sim	0.3196	0.0256	0.0460
	bigram, lda_sim	0.5480	0.1095	***0.1810***

Table 3. Fixed and Selected Feature Sets for SVM (RBF) and their Precision, Recall, F_1

Fixed Features	Selected Features	P	R	F_1
		0.5720	0.1774	***0.2679***

tfidf_sim, idf_sim, jaccard_sim	ssid	0.3063	0.1622	0.2091
	lda_sim	0.6221	0.1510	***0.2411***
	jaccard_sim	0.5924	0.1550	***0.2438***
	ssid, lda_sim	0.3450	0.1806	***0.2332***
	ssid, jaccard_sim	0.3224	0.1462	0.2002
	lda_sim, jaccard_sim	0.5579	0.1358	0.2166
	ssid, lda_sim, jaccard_sim	0.3594	0.1822	***0.2373***

Table 4. Fixed and Selected Feature Sets for LR and their Precision, Recall, F_1

Fixed Features	Selected Features	P	R	F_1
tfidf_sim, idf_sim, jaccard_sim		0.6230	0.1805	0.2787
	sec_position	0.6357	0.1885	0.2869
	lda_sim	0.6225	0.1845	0.2827
	sec_position, lda_sim	0.6375	0.2036	0.3060

Table 5. Fixed and Selected Feature Sets for DT and their Precision, Recall, F_1

Fixed Features	Selected Features	P	R	F_1
tfidf_sim, idf_sim, jaccard_sim, sent_position, sid		0.4131	0.4098	0.4102
	inner_position	0.3840	0.3843	0.3877
	lda_sim	0.3976	0.3779	0.3880
	d2v_sim	0.3512	0.3659	0.3616
	w2v_sim	0.3863	0.3923	0.3776
	inner_position, lda_sim	0.3974	0.3746	0.3866
	inner_position, d2v_sim	0.4004	0.3730	0.3811
	inner_position, w2v_sim	0.4170	0.4139	0.4152
	lda_sim, d2v_sim	0.3646	0.3819	0.3632
	lda_sim, w2v_sim	0.3843	0.3802	0.3826
	d2v_sim, w2v_sim	0.3574	0.3635	0.3518
	inner_position, lda_sim, d2v_sim	0.3792	0.3786	0.3752
	inner_position, lda_sim, w2v_sim	0.4060	0.4122	0.3963
	inner_position, d2v_sim, w2v_sim	0.3709	0.3707	0.3706
	lda_sim, d2v_sim, w2v_sim	0.3818	0.4066	0.3815
	inner_position, lda_sim, d2v_sim, w2v_sim	0.3858	0.3794	0.3730

As we can see, Decision Tree and Logistic Regression are performing better than SVM (Linear and RBF). Therefore, when doing integrations over classifiers, we construct two voting system, one is 4-classiferis containing all classifiers, another one is 3-classifiers where we remove the SVM (Linear).

Parameter Setting. In this system, voting weights of multi-classifiers and running setting are important parameters to adjust. Based on Table 2 to Table 5, we compute the average of precision, recall, F_1 for each classifier and use these average values as the voting system weights. Since the SVM (Linear) behave worst among all four systems, we do another voting system which only based on the other three classifiers. Voting weights for 4-classifiers and 3-classifiers are shown in Table 6 and Table 7.

Table 6. Different Voting Weights of Precision, Recall and F_1-Oriented 4-Classifiers System

Voting System	Classifiers	Voting Weight	Voting System	Classifiers	Voting Weight	Voting System	Classifiers	Voting Weight
Precision-oriented	SVM (Linear)	0.2160	Recall-oriented	SVM (Linear)	0.0699	F_1- oriented	SVM (Linear)	0.0954
	SVM (RBF)	0.2443		SVM (RBF)	0.2039		SVM (RBF)	0.2320
	DT	0.2051		DT	0.4870		DT	0.3829
	LG	0.3346		LG	0.2392		LG	0.2897

Table 7. Different Voting Weights of Precision, Recall and F_1-Oriented 3-Classifiers System

Voting System	Classifiers	Voting Weight	Voting System	Classifiers	Voting Weight	Voting System	Classifiers	Voting Weight
Precision-oriented	SVM (RBF)	0.3116	Recall-oriented	SVM (RBF)	0.2192	F_1- oriented	SVM (RBF)	0.2565
	DT	0.2617		DT	0.5236		DT	0.4233
	LG	0.4268		LG	0.2572		LG	0.3202

New Classifier. Except the classifiers we applied before, we also utilize a new one, called XGBOOST, which is an efficient and scalable implementation of gradient boosting framework by [21]. We use it as a single classifier with integrating into the voting system. When testing on training data, we select negative and positive samples in: 2, 3, 4 and 5. Figure 3 shows the average F_1.

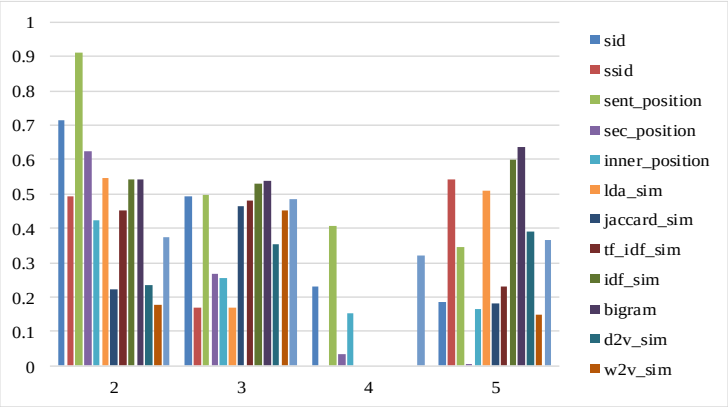


Fig. 4. Average F_1 of All Features with XGBOOST

Therefore, we also choose the fixed feature (bigram, IDF similarity and WordNet similarity) and selected feature sets (LDA similarity and Doc2Vec similarity) for XGBOOST and test again on training data when negative/positive samples, penalty factor are 5.5, 6, 6.5 and 7. Their performance of F_1 are show in Table 8 below.

Table 8. Fixed and Selected Feature Sets for XGBOOST and their F_1

Fixed Features	Selected Features	5.5	6	6.5	7
bigram, idf_sim, wordnet_simi		***0.5746***	0.5647	0.4931	0.4647
	lda_sim	***0.6231***	0.5562	0.5309	0.4974
	d2v_sim	***0.5868***	0.4846	0.5212	0.4252
	d2v_sim,lda_sim	***0.7316***	0.5588	0.5740	0.5123

3.2 Task 1B

In this task, for each cited text span, we need to identify what facet of the paper it belongs to. Basically, there are three components in this system to deal with Task 1B.

- Dictionary: We construct two kinds of dictionaries of five facets manual dictionary, and POS dictionary. The first one is made manually and latter one is made according to part-of-speech tagging results. For POS dictionary, we keep those words whose POS results are VB and JJ. In detail, method POS dictionary has words which frequency is over 5, and for the other facet POS dictionary, they has words which frequency is over 2.
- Supervised Topic Model: After proposing of latent sematic indexing, latent topic modeling has become very popular for topic discovery in document collections, such as Latent Dirichlet Allocation (LDA) [22]. Supervised topic model (LLDA) [23] is then followed by, which can overcome limitations of traditional ones. This model assumes availability of topic labels (keywords) and the characterization of each topic by a multinomial distribution over all vocabulary words.
- XGBOOST: Tree boosting is a highly effective and widely used machine learning method. Here we apply XGBOOST [24] for approximate tree learning. When training the model, there are 15 features in total. Five of them are the matched word number based on manual dictionary, five of them are the matched word number based on POS dictionary, and the left ones are position-based features mentioned in section 3.1.

Based on three components above, there are five different strategies:

Manual Dictionary. Based on the five different dictionaries of five facets, if the section title or sentence content contains any one of these words in the corresponding built dictionaries, it will be directly classified as the corresponding facet. Since the manual dictionary will be more accurate than POS dictionary. We only apply this strategy using manual dictionary. When doing judgements, the first identified facet should contain more than $1(Count_{M1})$ word in dictionary, the second identified facet should contain more than $2(Count_{M2})$ words in dictionary. To find the best order of judging facets, we do the experiments over all random arrangements. In total, there are 120 sets of results, here we only show the top 20 ones based on F_1 in Table 9.

Table 9. Top 20 Average F_1 Generated via Different Judging Orders Using Manual Dictionary

Judging Order	F_1	Judging Order	F_1
implication->method->result->aim->hypothesis	0.7179	method->result->hypothesis->implication->aim	0.7159
implication->method->result->hypothesis->aim	0.7179	method->result->hypothesis->aim->implication	0.7159
implication->method->hypothesis->result->aim	0.7179	method->hypothesis->result->implication->aim	0.7159
implication->method->aim->result->hypothesis	0.7162	method->hypothesis->result->aim->implication	0.7159
implication->method->aim->hypothesis->result	0.7162	implication->hypothesis->method->result->aim	0.7146
implication->method->hypothesis->aim->result	0.7162	hypothesis->implication->method->result->aim	0.7146
method->result->implication->aim->hypothesis	0.7159	method->implication->result->aim->hypothesis	0.7146
method->result->implication->hypothesis->aim	0.7159	method->implication->result->hypothesis->aim	0.7146
method->result->aim->implication->hypothesis	0.7159	method->implication->hypothesis->result->aim	0.7146
method->result->aim->hypothesis->implication	0.7159	method->hypothesis->implication->result->aim	0.7146

LLDA. For training data, we assume that each identified facet is a topic label and that each citation sentence is a mixture of the expert-assigned topics that can be learned. We firstly trained LLDA model on the training data and the dimension number is five. Then, we apply this trained model to do predictions over testing data. Here, there is no labels for testing data yet. After representing each sentence into the probability distribution over five facets, we recognize the most possible facet as its identified facet. Since some sentences might have more than one facets, we set the possibility thresholds ($P_{LLDA2} = 0.2\ or\ 0.195$) for the second possible facet. Referring to LLDA parameters, we do some adjustments on beta, where a low beta value places more weight on having each topic composed of only a few dominant words. Table 10 shows different beta settings and their corresponding F_1.

Table 10. Average F_1 under Different Beta Settings

Beta	F_1	Beta	F_1	Beta	F_1	Beta	F_1
0.1	0.3576	0.5	0.6939	1.2	0.7278	2	0.7228
0.2	0.5005	0.7	0.723	1.5	0.7241	5	0.7228

XGBOOST. Here, we use the XGBOOST to do classification in this task. When choosing features, position-based features mentioned in section 3.1 are selected as selected feature set which will be evaluated using its candidate subsets. Performance of different selected feature sets are given below in Table 11.

Table 11. Selected Feature Sets for XGBOOST and their F_1

Selected Feature Set	F_1	Selected Feature Set	F_1
sid, sid_position	0.7114	sid, ssid, sid_position, section_position	0.7039
sid, inner_position	0.7102	sid_position, section_position	0.7029
sid	0.7077	sid, ssid, inner_position, section_position	0.7027
sid, sid_position, inner_position, section_position	0.7077	sid, ssid, sid_position, inner_position, section_position	0.7014
sid_position, inner_position	0.7065	ssid, sid_position	0.7004
sid, sid_position, inner_position	0.7065	ssid, sid_position, section_position	0.7004
sid_position	0.7054	sid, ssid, sid_position, inner_position	0.7003
ssid, sid_position, inner_position	0.7053	inner_position	0.7002
inner_position, section_position	0.7052	ssid	0.6992
sid, ssid, sid_position	0.7052	section_position	0.6992
sid, ssid, inner_position	0.7052	sid, ssid	0.699
sid, inner_position, section_position	0.7052	sid, ssid, section_position	0.699
sid_position, inner_position, section_position	0.7052	ssid, inner_position, section_position	0.699
sid, sid_position, section_position	0.705	ssid, sid_position, inner_position, section_position	0.699
sid, section_position	0.7039	ssid, section_position	0.6979
ssid, inner_position	0.6978		

Manual dictionary + LLDA. Different from LLDA strategy, we use the manual dictionary-labeled testing data to be the testing data for LLDA prediction. Here, we also set the possibility thresholds for the second possible facet ($P_{LLDA2} = 0.18$) and the thresholds for contained word counts of the first and second identified facet when doing different order of judgements ($Count_{M1} = 1$ and $Count_{M2} = 2$) To find the best order of judging facets, we also do the experiments over all random arrangements. Here we only show the top 20 ones based on F_1 in table 12.

Table 12. Top 20 Average F_1 Generated via Different Judging Orders

Judging Order	F_1	Judging Order	F_1
implication->method->result->aim->hypothesis	0.7191	method->result->hypothesis->implication->aim	0.7165
implication->method->result->hypothesis->aim	0.7191	method->result->hypothesis->aim->implication	0.7165
implication->method->hypothesis->result->aim	0.7191	method->hypothesis->result->implication->aim	0.7165
implication->method->aim->result->hypothesis	0.7178	method->hypothesis->result->aim->implication	0.7165
implication->method->aim->hypothesis->result	0.7178	implication->hypothesis->method->result->aim	0.7157
implication->method->hypothesis->aim->result	0.7178	hypothesis->implication->method->result->aim	0.7157
method->result->implication->aim->hypothesis	0.7165	method->aim->result->implication->hypothesis	0.7152
method->result->implication->hypothesis->aim	0.7165	method->aim->result->hypothesis->implication	0.7152
method->result->aim->implication->hypothesis	0.7165	method->aim->hypothesis->result->implication	0.7152
method->result->aim->hypothesis->implication	0.7165	method->hypothesis->aim->result->implication	0.7152

POS dictionary + LLDA. Similar with previous method, we use the POS dictionary-labeled testing data to be the testing data for LLDA prediction. We also set the same three parameters in this strategy, where $P_{LLDA2} = 0.18$, $Count_{P1} = 3$ and $Count_{P2} = 8$. The top 20 F_1 via different judging order is give in Table 13.

Table 13. Top 20 Average F_1 Generated via Different Judging Orders

Judging Order	F_1	Judging Order	F_1
method->implication->result->aim->hypothesis	0.7511	method->implication->aim->result->hypothesis	0.7498
method->implication->result->hypothesis->aim	0.7511	method->implication->aim->hypothesis->result	0.7498
method->implication->hypothesis->result->aim	0.7511	method->implication->hypothesis->aim->result	0.7498
method->hypothesis->implication->result->aim	0.7511	method->aim->implication->result->hypothesis	0.7498
method->result->implication->aim->hypothesis	0.7498	method->aim->implication->hypothesis->result	0.7498
method->result->implication->hypothesis->aim	0.7498	method->aim->hypothesis->implication->result	0.7498
method->result->aim->implication->hypothesis	0.7498	method->hypothesis->result->implication->aim	0.7498
method->result->aim->hypothesis->implication	0.7498	method->hypothesis->result->aim->implication	0.7498
method->result->hypothesis->implication->aim	0.7498	method->hypothesis->implication->aim->result	0.7498
method->result->hypothesis->aim->implication	0.7498	method->hypothesis->aim->implication->result	0.7498

3.3 Task 2

Summary generation is divided into two main steps. First is to group sentences into different clusters based on its similarity with different parts of abstract. Second is using several features to extract sentence from each cluster and combine them into a summary.

Normally, abstract is a complete but concise description of the work. In particular, different parts may be merged or spread among a set of sentences, like motivation, problem statement, approach, results and conclusions. Therefore, we wants to organize the abstract sentences of reference paper in advance, and group the identified cited spans based on their similarities between different parts of abstract sentences. Basically, we assume that abstract will contain motivation, approach and conclusion. In order to split them into these three group, we apply rule-based method based on writing styles. We find that when people write summaries like abstract, they will start with some fixed phrases, such as “this paper”, “in this paper” or “we”. If the first sentence doesn’t have

these fixed phrases, it will be about motivation of this paper for most of the time. Meanwhile, the last sentence are usually about results or conclusions.

Therefore, we firstly split abstract sentences into groups if they follow these rules. Then, each identified text span is selected into different groups based on their similarity with the grouped abstract sentences. Here we use the linear sum of Jaccard, IDF and TFIDF similarities. After this, we rank the sentences within each group, using weighted features of those three similarities, sentence length and sentence position. Formula is shown below:

$$Score_i = 2.5S_{Jaccard} + 2.5S_{IDF} + 2.5S_{TFIDF} + 1.25S_{Length} + 1.25S_{Position} \quad (1)$$

Finally, for each time, we choose first one sentence from each cluster to build the summary before the length of summary exceeds 250 words.

4 Experiments

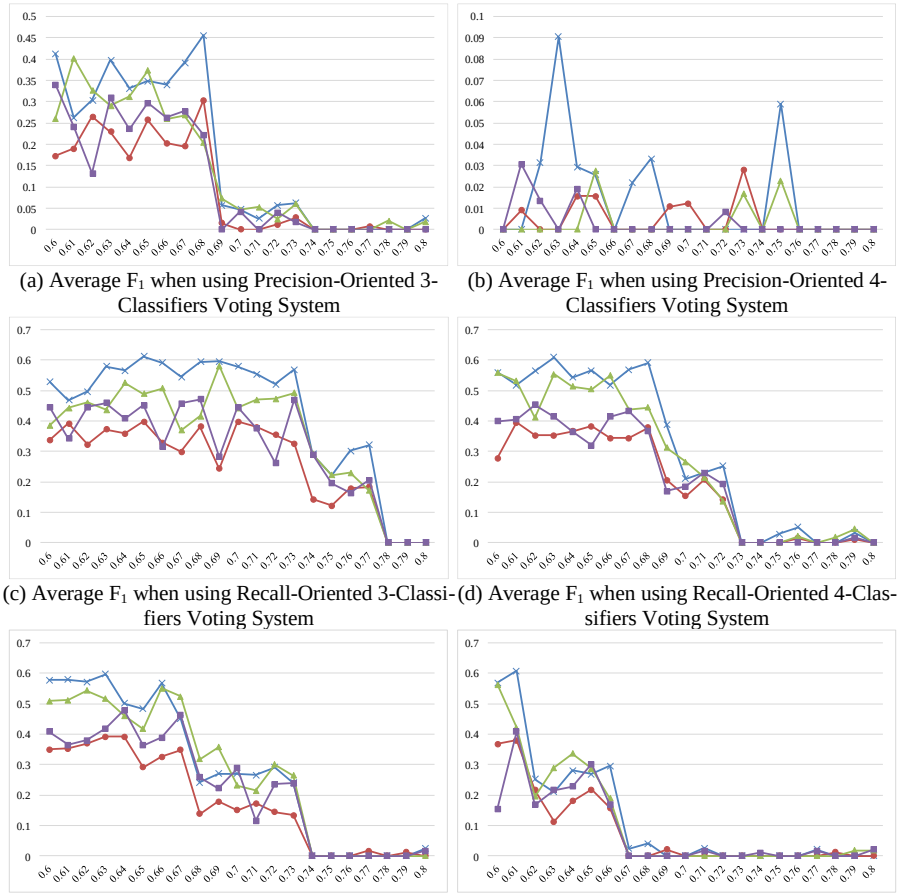
4.1 Data and Tools

When doing corpora preprocessing, we remove the stop words and stem words to base forms by Porter Stemmer algorithm[6]. Then, we applied Word2Vec and Doc2Vec model in Genism[7] and python package of LDA[8] model to represent documents. All the classifiers were done via Scikit-learn python package[9]. XGBOOST is obtained via a python extension package website[10]. Source code of our system will be made available at: https://github.com/michellemashutian/NJUST-at-CLSciSumm/tree/master/NJUST-2018.

4.2 Submission Results

Task 1A. After using the best feature combinations on 4-classifiers and 3-classifiers, testing on different parameters, we obtain the average F_1 shown in Figure 4. Proportion of negative/positive samples, penalty factor are tested on 5.5(blue cross line), 6 (red circle line), 6.5 (green triangle line) and 7 (purple square line). Thresholds range from 0.6 to 0.8, as 0.01 is the interval (x axis).

[6] Available at: http://snowball.tartarus.org/algorithms/porter/stemmer.html
[7] Available at: https://radimrehurek.com/gensim/
[8] Available at: https://pypi.org/project/lda/
[9] Available at: http://scikit-learn.org/stable/index.html
[10] Available at: https://www.lfd.uci.edu/~gohlke/pythonlibs/#xgboost



(a) Average F_1 when using Precision-Oriented 3-Classifiers Voting System

(b) Average F_1 when using Precision-Oriented 4-Classifiers Voting System

(c) Average F_1 when using Recall-Oriented 3-Classifiers Voting System

(d) Average F_1 when using Recall-Oriented 4-Classifiers Voting System

(e) Average F_1 when using F_1-Oriented 3-Classifiers Voting System

(f) Average F_1 when using F_1-Oriented 4-Classifiers Voting System

Fig. 5. Average F1 when using the Best Feature Combinations on 4-classifiers and 3-classifiers

According to Figure 4, we pick the Top 10 performance of multi-classifiers and their parameters are given in Table 14. Except voting system, we also submit another 10 running results which are obtained via single classifiers. Parameter and classifier features are given in Table 15.

Table 14. Parameter Settings for Task 1A Submissions Using Voting System.

Voting System	Voting Weights	#Neg/#Pos Penalty Factor	Thresholds	Voting System	Voting Weights	#Neg/#Pos Penalty Factor	Thresholds
3 Classifiers	Precision	5.5	0.68	4 Classifiers	Precision	5.5	0.63
	Recall	5.5	0.65		Recall	5.5	0.63
	F_1	5.5	0.61/0.63		F_1	5.5/6.5	0.6/0.61

Table 15. Parameter Settings for Task 1A Submissions Using Single Classifier.

Classifiers	#Neg/#Pos Penalty Factor	Features

DT	5.5	tf_idf_sim,idf_sim,jaccard_sim,sent_position,sid, inner_position,w2v_sim
DT	5.5	tf_idf_sim,idf_sim,jaccard_sim,sent_position,sid, None
LG	5.5	tf_idf_sim,idf_sim,jaccard_sim, sec_position,lda_sim
LG	5.5	tf_idf_sim,idf_sim,jaccard_sim, sec_position
SVM(RBF)	5.5	tf_idf_sim,idf_sim,sid, jaccard_sim
SVM(RBF)	5.5	tf_idf_sim,idf_sim,sid
XGBOOST	5.5	bigram,idf_sim,wordnet_sim, d2v_sim,lda_sim
XGBOOST	5.5	bigram,idf_sim,wordnet_sim, d2v_sim
XGBOOST	5.5	bigram,idf_sim,wordnet_sim, lda_sim
XGBOOST	5.5	bigram,idf_sim,wordnet_sim

Task 1B. Referring the five strategies using dictionary, based on the performance of different judgment order (Table 9, Table 12 and Table 13), we select the specific order according to their F_1 results, when they generate the same facet identification on testing data, we just move to next order which has lower F_1. For LLDA strategy, we pick the top 4 results with corresponding beta settings to run on test data. For XGBOOST strategy, we also select top 4 results with corresponding feature selections to run on test data. Table 16 shows the overall parameter settings of our Task 1B submission.

Table 16. Parameter Settings for Task 1B Submissions Using Five Strategies.

Strategy	Parameter Setting
LLDA	$\beta_{LLDA} = 1.2, P_{LLDA2} = 0.2$
	$\beta_{LLDA} = 1.2, P_{LLDA2} = 0.195$
	$\beta_{LLDA} = 1.5, P_{LLDA2} = 0.2$
	$\beta_{LLDA} = 1.5, P_{LLDA2} = 0.195$
Manual Dictionary	implication->hypothesis->method->result->aim, $Count_{M1} = 1$ and $Count_{M2} = 2$
	implication->method->result->aim->hypothesis, $Count_{M1} = 1$ and $Count_{M2} = 2$
	method->result->implication->aim->hypothesis, $Count_{M1} = 1$ and $Count_{M2} = 2$
Manual Dictionary+LLDA	$\beta_{LLDA} = 1.2, P_{LLDA2} = 0.18$, implication->method->result->aim->hypothesis, $Count_{M1} = 1$ and $Count_{P2} = 2$
	$\beta_{LLDA} = 1.2, P_{LLDA2} = 0.18$, method->result->implication->aim->hypothesis, $Count_{M1} = 1$ and $Count_{P2} = 2$
	$\beta_{LLDA} = 1.2, P_{LLDA2} = 0.18$, implication->method->result->aim->hypothesis, $Count_{P1} = 1$ and $Count_{P2} = 3$
POS Dictionary+LLDA	$\beta_{LLDA} = 1.2, P_{LLDA2} = 0.18$, method->implication->aim->result->hypothesis, $Count_{P1} = 3$ and $Count_{P2} = 8$
	$\beta_{LLDA} = 1.2, P_{LLDA2} = 0.18$, method->implication->result->aim->hypothesis, $Count_{P1} = 3$ and $Count_{P2} = 8$
	$\beta_{LLDA} = 1.2, P_{LLDA2} = 0.18$, method->result->implication->aim->hypothesis, $Count_{P1} = 3$ and $Count_{P2} = 8$
XGBOOST	sid, sid_position
	sid, inner_position
	sid
	sid, sid_position, inner_position, section_position

5 Conclusion

This document demonstrates our participant system NJUST on CL-SciSumm 2018. Compared with previous system, we has added some semantic information like WordNet and Word2Vec similarities to improve the citance linkage and summarization per-

formance. We also optimize the bigram feature. When choosing feature and setting parameters, comparative experiments are finished systematically. New methods are proposed in this paper to deal with facet identification and automatic summarizations. In Task 1B, rule-based methods are combined with supervised topic modeling and XGBOOST. As to Task 2, we take advantages of abstract structures.

In the future work, more things can be done on these three tasks. For Task 1A and Task 1B, we can try new classifiers to see the performance. For Task 2, we need to find more features to calculate the sentence score for ranking, such as sentence position, etc. We can also make use of the results in Task 1B to generate a more reasonable summary.

Acknowledgements

This work is supported by Major Projects of National Social Science Fund (No. 17ZDA291), Fujian Provincial Key Laboratory of Information Processing and Intelligent Control (Minjiang University) (No. MJUKF201704) and Qing Lan Project.

References

1. Stevenson, S. and E. Joanis. *Semi-supervised verb class discovery using noisy features*. in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. 2003. Association for Computational Linguistics.
2. Chieu, H.L. and H.T. Ng. *Named entity recognition: a maximum entropy approach using global information*. in *Proceedings of the 19th international conference on Computational linguistics-Volume 1*. 2002. Association for Computational Linguistics.
3. Qazvinian, V., et al., *Generating extractive summaries of scientific paradigms.* Journal of Artificial Intelligence Research, 2013. **46**: p. 165-201.
4. Waard, A.d. and H.P. Maat, *Epistemic modality and knowledge attribution in scientific discourse: a taxonomy of types and overview of features*, in *Proceedings of the Workshop on Detecting Structure in Scholarly Discourse*. 2012, Association for Computational Linguistics: Jeju, Republic of Korea. p. 47-55.
5. Ma, S., et al. *NJUST@ CLSciSumm-17*. in *Proc. of the 2nd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL2017). Tokyo, Japan (August 2017)*. 2017.
6. Ma, S., J. Xu, and C. Zhang, *Automatic identification of cited text spans: a multi-classifier approach over imbalanced dataset.* Scientometrics, 2018.
7. Ware, M. and M. Mabe, *The STM report: An overview of scientific and scholarly journal publishing.* 2015.
8. Jaidka, K., et al., *Insights from CL-SciSumm 2016: the faceted scientific document summarization Shared Task.* International Journal on Digital Libraries, 2017: p. 1-9.
9. Jaidka, K., et al. *The CL-SciSumm shared task 2017: results and key insights*. in *Proceedings of the Computational Linguistics Scientific Summarization Shared Task (CL-SciSumm 2017), organized as a part of the 2nd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2017)*. 2017.

10. Li, L., et al. *CIST System for CL-SciSumm 2016 Shared Task*. in *BIRNDL@ JCDL*. 2016.
11. Aggarwal, P. and R. Sharma. *Lexical and Syntactic cues to identify Reference Scope of Citance*. in *BIRNDL@ JCDL*. 2016.
12. Cao, Z., W. Li, and D. Wu. *PolyU at CL-SciSumm 2016*. in *BIRNDL@ JCDL*. 2016.
13. Prasad, A. *WING-NUS at CL-SciSumm 2017: Learning from Syntactic and Semantic Similarity for Citation Contextualization*. in *Proc. of the 2nd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL2017). Tokyo, Japan (August 2017)*. 2017.
14. Li, L., et al. *CIST@ CLSciSumm-17: Multiple Features Based Citation Linkage, Classification and Summarization*. in *Proc. of the 2nd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL2017). Tokyo, Japan (August 2017)*. 2017.
15. Abura'ed, A., et al., *LaSTUS/TALN@ CLSciSumm-17: cross-document sentence matching and scientific text summarization systems.* 2017.
16. Moraes, L., et al. *University of Houston at CL-SciSumm 2016: SVMs with tree kernels and Sentence Similarity*. in *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL)*. 2016.
17. Lauscher, A., G. Glavaš, and K. Eckert, *University of Mannheim@ CLSciSumm-17: Citation-Based Summarization of Scientific Articles Using Semantic Textual Similarity.* 2017.
18. Zhang, D. and S. Li. *PKU@ CLSciSumm-17: Citation Contextualization*. in *Proc. of the 2nd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL2017). Tokyo, Japan (August 2017)*. 2017.
19. Klampfl, S., A. Rexha, and R. Kern. *Identifying referenced text in scientific publications by summarisation and classification techniques*. in *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL)*. 2016.
20. Saggion, H. and F. Ronzano. *Trainable citation-enhanced summarization of scientific articles*. in *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL)*. 2016.
21. Friedman, J.H., *Greedy function approximation: a gradient boosting machine.* Annals of statistics, 2001: p. 1189-1232.
22. Blei, D.M., A.Y. Ng, and M.I. Jordan, *Latent dirichlet allocation.* Journal of machine Learning research, 2003. **3**(Jan): p. 993-1022.
23. Mcauliffe, J.D. and D.M. Blei. *Supervised topic models*. in *Advances in neural information processing systems*. 2008.
24. Chen, T. and C. Guestrin. *Xgboost: A scalable tree boosting system*. in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016. ACM.