Team MLU@CL-SciSumm20: Methods for Computational Linguistics Scientific Citation Linkage

Rong Huang, Kseniia Krylova
Martin-Luther-Universität Halle-Wittenberg (MLU)

Abstract

This paper describes our approach to the CL-SciSumm 2020 shared task toward the problem of identifying reference span of the citing article in the referred article. In Task 1a, we apply and compare different methods in combination with similarity scores to identify spans of the reference text for the given citance. In Task 1b, we use a logistic regression to classifying the discourse facets.

1 Introduction

The CL-SciSumm Shared Task focuses on automatic paper summarization in the domain of computational linguistics research. Given a document set with a reference papers and citing papers that all contain citations to the reference paper. In Task 1a we should identify the spans of text (cited text spans) in the reference paper that most accurately reflect the citance. In Task 1b for each cited text span, we should identify what facet of the paper it belongs to, from a predefined set of facets: hypothesis, aim, method, results, and implication. Task 2 is to generate a summary of the reference paper. In this work, we focus on Task 1.

For comparison purposes, we experimented with the following approaches: SVM, logistic regression, decision tree (CART), voting, and calculated a set of similarity metrics between reference spans and citance: tf-idf approach, cosine similarity, Jaccard similarity, WordNet similarity. The best results are obtained by method which combined similarity scores using tf-idf approach, cosine similarity, WordNet similarity, bigram distance and SVM.

We also analyzed a dataset to identify features and highlights, which help us in solving the task. We found that most reference spans contain only one sentence. Also, we conduct a semantic analysis to extract the named entities such as persons, organizations, products or locations. The most common named entities such as organizations and persons can be used for feature extraction.

2 Dataset

The dataset contains 40 topics with citation sentences and human-annotated reference summaries. Each topic is composed of a Reference Paper (RP) and some Citing Papers (CP). We have separate the data into two sets: 30 for training and 10 for testing. For this analysis all uppercase letters in train dataset were transformed into lowercase letters and all words that included non-alphabetical characters were removed. There are 46451 unique words among the reference documents. There are 648 reference sentences and 559 citances. An average number of reference sentences is 1.2, that means, a citation text can be linked to many sentences in the reference paper. The Fig. 1 shows the distribution of the number of sentence in reference spans. The horizontal axis represents the

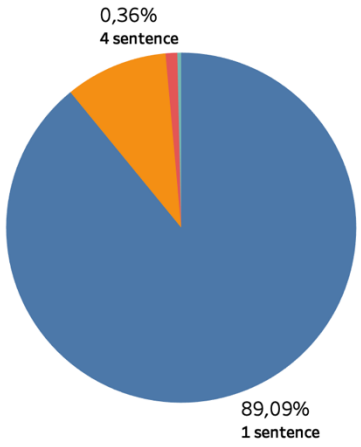

Figure 1 A distribution of the number of sentence in reference spans for train dataset

Proceedings of the First Workshop on Scholarly Document Processing, pages 282–287

https://doi.org/10.18653/v1/P17

number of sentences in each reference span. The vertical axis shows the frequency. As can be seen, a large proportion of reference spans (about 90%) in train dataset contain only one sentence. We assume to fit our model in such a way it identifies one reference sentence referred to by a given citance.

Citations play an important role in understanding the relationship between scientific works that are related to each other (Iman Tahamtan et al., 2019). Given the citation texts, we find the text spans in the reference article that most closely reflect the citation text.

The noteworthy feature here is that citances have a few peculiarities, such as an abundance of citation markers and proper names. Citations sometimes include the names of the authors, which results in more frequent use of our own proposals. In our work we identify and ignore citation markers such as the author's name, which allows to reduce noise.

We propose syntactic and semantic analysis of citation content that can be used to better analyze the context of research behavior. There are 717 entities for 559 citances. Table 1 shows entity frequency classified by entity types.

We considered the distribution of the following elements: organizations (companies, agencies, institutions), persons (people, including fictional), products (objects, vehicles, foods etc.) and locations (countries, cities, states). The total number of tokens in reference spans 9076.

Entity type	Count	Description
ORG	269	Companies, agencies, institutions
PERSON	157	People, including fictional
PRODUCT	41	Objects, vehicles, foods, etc. (Not services.)
GPE	21	Countries, cities, states

Table 1: Entity distribution classified by entity

The distribution of entities and percentage are presented in Fig.2. The horizontal axis represents one of entities types, which we are considering. The vertical axis shows the frequency. As can be seen from the figure, the most citied are organizations – 37,52% of the number of citied entities (total 269). It represents 2,96% of the total number of tokens in reference spans. This is followed by persons, which represent 21,9% of the total number of entities and 1,73% of total number of tokens. Products make up 5,72% of number of citied entities and location represent nearly 3% of total number of citied entities.

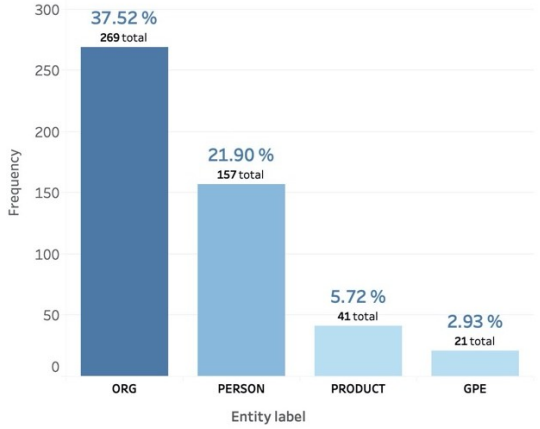

Figure 2: Entity distribution classified by entity types in reference spans in train dataset.

Based on this distribution, we choose first three most common named entities – organizations (ORG), persons (PERSON) and products (PRODUCT) to find a number of matches between named entities in citation and a citation-candidate in a reference text.

3 Approach

3.1 Task 1a

In Task 1a we should identify cited text spans in the reference paper for each citance. Applied methods are highly relevant to the methods of calculating similarity. We propose different approaches that complement each other.

The first approach based on tf-idf cosine similarity between a citance and a reference span in the reference paper. In this approach we transform our work into binary classification problem which is to classify every sentence in the reference paper into relevant or irrelevant. Each reference sentence is assigned a score according to the cosine similarity between tf-idf vectors of itself and the citance. The major feature we use is the threshold, which is manually selected based on analysis of similarity and our experiments.

We consider Task 1a as a classification task, which is to find reference span by a given citance. In this case feature selection plays an important role in identifying citances from reference. It is proposed four categories of features: location-

based features, sentence importance-based features, similarity-based features and rule-based features (Qi Zhang et al., 2019). Location-based features contain the information about position of the sentence in the reference paper. Similarity-based features indicate similarity measures between the citance and the reference sentence. Finally, rule-based features refer to identifying citances from reference paper using manual rules. We choose following features:

Sentence Position (SID). We choose the serial number of a citance candidate (sid) in a full reference text as a location-based feature.

Jaccard similarity (JS). Jaccard similarity coefficient is a statistic used for gauging the similarity and diversity of sample sets. It is defined as the size of the intersection divided by the size of the union of two sets. We choose JS as one of the similarity-based features.

Bigram distance (BD). Another similarity-based feature work by converting strings into sets of n-grams. The similarity or distance between the strings is then the similarity or distance between the sets. For this purpose, we used a set class that supports lookup by N-gram string similarity provided by NGram Module.

Count Cosine Similarity (CS). Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space. We use CountVectorizer from scikit-learn to convert sentences into vectors.

Tf-Idf Cosine Similarity (TFIDF). The difference of this method compared to the previous one (CS) is using of TfIdfVectorizer instead of CountVectorizer.

WordNet Similarity (WS). WordNet is a lexical database for the english language (Miller, 1995). Synonymous words are grouped into sets of cognitive synonyms (synsets), each expressing the same concept. Synsents are organized in a structure similar to inheritance tree. More abstract words called hypernyms and more specific are hyponyms. This tree can be used for calculating similarity between two sentences. The closer the two Synsets are in the tree, the more similar they are (Nitin Hardeniya et al., 2016). For this purpose, we use WordNet provided by NLTK package. This algorithm was proposed by Mihalcea et al. (2006).

Matches Named Entities (ME). We find the number of matches between named entities in citation and a citation-candidate in a reference text. Based on semantic analysis, we choose organizations (ORG), persons (PERSON) and products (PRODUCT).

Matches Named Entities Labels (MEL): We find the number of matches between labels of named entities in citation and a citation-candidate in a reference text. This features are defined and set manually, therefore they are rule-based features.

General Inquirer Category Listings (INQ). General Inquirer Category Listings is a dictionary that contains it about 12 000 words, divided into categories. Each category is a list of words and word senses (Stone, 2006). We indicate whether there are words from General Inquirer Category Listings both in citance and in reference sentence. It is also a rule-based feature.

General Inquirer Category Listings - Sentiment (INQS). There are two large valence categories - 1,915 words of positive outlook and 2,291 words of negative outlook. Each sentence pair is assigned a score by taking scores for each token by using positive or negative labels.

After we defined features, we classify the pairs of citance and reference sentence as relevant or irrelevant.

Logistic regression (LR). We use similarity-based features and rule-based features with logistic regression to classify sentences as being reference spans or not.

Voting (VT). Voting classifier is a machine learning model that trains on a collection of fitted sub-estimators. The predicted output class is a class with the highest majority of votes i.e. the class which had the highest probability of being predicted by each of the classifiers. We defined as sub-estimators logistic regression, SVM and Decision Tree Classifier.

3.2 Task 1b

For Task 1b, we use a logistic regression with bag-of-words as features.

For Task 1, the distribution of examples across the classes is not equal and it makes the problem strongly imbalanced. In our case, class "irrelevant" is present with 15:1 ratio in training set. To solve the problem of imbalance, we use classifier SVM with stochastic gradient descent (SGD) training. For this purpose, we use SGDClassifier provided by scikit-learn, which yield behavior such as that of a SVC with a linear kernel for classes that are unbalanced (Pedregosa et al., 2011).

4 Experiment

The dataset Training-Set-2018 provided by CL-SciSumm Shared Task are training data and test data in our system. The dataset contains 40 topics with citation sentences and human-annotated reference summaries. As described in section "Dataset", we separated data into two sets: 30 for training and 10 for testing. The documents were selected in alphabetical order.

Before we use dataset in our system we preprocessed the dataset to reduce some xml-coding errors. Formatting problems such a missed tags, broken words, non-ascii characters in XML files are some examples of these problems. We manually fixed broken words and automatically removed all non-ascii characters. We also fixed some missed closing tags.

To remove the effect of using words to their different words we used lemmatization. This process is used to return the word to its origin. Stopwords were removed for all configurations.

We have also limited reference sentences by number of words. The average number of words in sentences is 47, minimal number is 6, maximal 282. In order to reduce noise we consider sentences with more than 10 and less than 70 words.

In our baseline method, first we analyzed results of similarity calculations in order to choose a threshold.

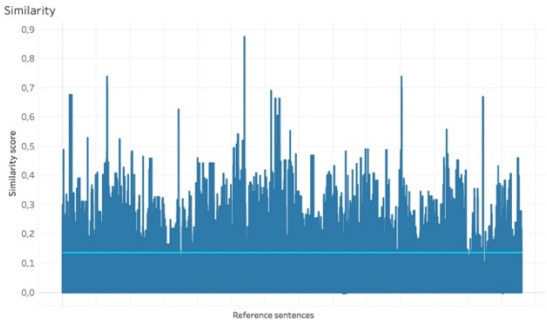

Figure 3: Demonstrates similarity score distribution. The horizontal axis represents reference sentences, the vertical axis represents similarity score. Based on our experimental results we defined threshold 0.14.

In approach method, we train our feature-based classifiers on all the relevant sentences pairs.

5 Results

We implement our system and use official scripts to evaluate the training data. The evaluation of our approaches is done by comparison of several metrics which are presented in tables below.

From Table 2, Table 3 and Table 4, we can find, that SVM method in combination with TF-IDF approach (TFIDF), Bigram Distance (BD), Jaccard Similarity (JS), Count Cosine Similarity (CS) and WordNet Similarity show better performance in our experiments.

The second best score is represented by SVM method in combination with TF-IDF approach (TFIDF), Bigram Distance (BD), Jaccard Similarity (JS), Count Cosine Similarity (CS), WordNet Similarity and Sentence Position (SID).

Method	Precision
TFIDF+Threshold	0.0507
SVM+TFIDF+BD+JS+CS	0.0648
SVM+TFIDF+BD+JS+CS+WS	**0.375**
SVM+SGD+TFIDF+BD+JS+CS+WS	0.5
SVM+SID+TFIDF+BD+JS+CS+WS	0.1389
SVM+SID+TFIDF+BD+JS+CS+WS+INQ+INQS	0.1111
SVM+TFIDF+BD+JS+CS+WS+ME+MEL	0.25
LR+TFIDF+BD+JS+CS	0.0806
LR+TFIDF+BD+JS+CS+WS	0.0694
LR+TFIDF+BD+JS+CS+WS+ME+ MEL	0.0139
DT+TFIDF+BD+JS+CS	0.0745
VT+TFIDF+BD+JS+CS	0.0926
VT+TFIDF+BD+JS+CS+WS	0.0556
VT+TFIDF+SID+BD+JS+CS+WS	0.0388

Table 2: Precision score metric

Our set of experimental results, shown in Figure 4, present the performance of baseline method and approach method with respect to the metrics of precision, recall and F1-score. The vertical axis represent metric values, the horizontal axis represent the evaluation metrics, namely precision, recall and F1-score.

Compares to the baseline, the increases in approach method are 640% (0.3750 vs 0.0507), 85% (0.0783 vs 0.0423) and 180% (0.1295 vs 0.0462) respectively.

However, performance degrades when we take into account data imbalance issue. Figure 5 demonstrates performance comparison for SVM approach method and for improved SVM method with stochastic gradient descent (SGD).

Method	**Recall**
TFIDF+Threshold	0.0423
SVM+TFIDF+BD+JS+CS	0.0469
SVM+TFIDF+BD+JS+CS+WS	**0.0783**
SVM+SGD+TFIDF+BD+JS+CS+WS	0.0505
SVM+SID+TFIDF+BD+JS+CS+WS	0.0863
SVM+SID+TFIDF+BD+JS+CS+WS+INQ+INQS	0.037
SVM+TFIDF+BD+JS+CS+WS+ME+MEL	0.0227
LR+TFIDF+BD+JS+CS	0.0513
LR+TFIDF+BD+JS+CS+WS	0.0528
LR+TFIDF+BD+JS+CS+WS+ME+ MEL	0.0139
DT+TFIDF+BD+JS+CS	0.0432
VT+TFIDF+BD+JS+CS	0.0570
VT+TFIDF+BD+JS+CS+WS	0.0370
VT+TFIDF+SID+BD+JS+CS+WS	0.0241

Table 3: Recall score metric

Method	**F1-score**
TFIDF+Threshold	0.0462
SVM+TFIDF+BD+JS+CS	0.0544
SVM+TFIDF+BD+JS+CS+WS	**0.1295**
SVM+SGD+TFIDF+BD+JS+CS+WS	0.0917
SVM+SID+TFIDF+BD+JS+CS+WS	0.1065
SVM+SID+TFIDF+BD+JS+CS+WS+INQ+INQS	0.0556
SVM+TFIDF+BD+JS+CS+WS+ME+MEL	0.0417
LR+TFIDF+BD+JS+CS	0.0627
LR+TFIDF+BD+JS+CS+WS	0.06
LR+TFIDF+BD+JS+CS+WS+ME+ MEL	0.0139
DT+TFIDF+BD+JS+CS	0.0574
VT+TFIDF+BD+JS+CS	0.0706
VT+TFIDF+BD+JS+CS+WS	0.0444
VT+TFIDF+SID+BD+JS+CS+WS	0.0297

Table 4: F1 score metric

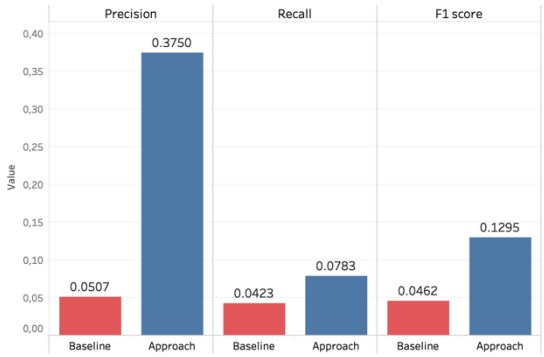

Figure 4: Comparison of performances of baseline method and approach

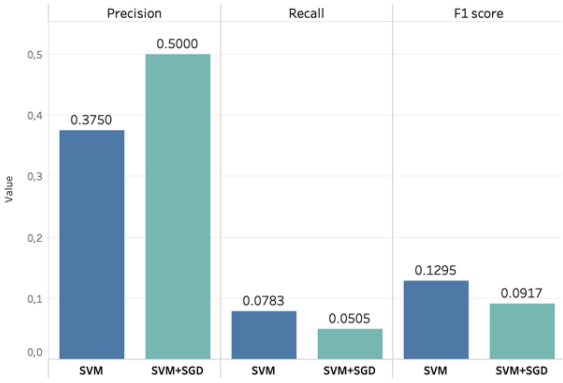

Figure 5: Performance comparison for SVM approach method and SVM approach with stochastic gradient descent (SGD)

6 Conclusion and future work

In this paper, we transform our work into binary classification problem and apply various methods to identify the spans of text in the reference paper reflecting the citance. We compare feature-based classifiers in combination with different features. Although results show an improvement over the baseline, it is important to improve the performance of imbalanced data classification.

Acknowledgements

We are very grateful to Professor Matthias Hagen and Mr. Yamen Ajjour for their guidance on our research work and programming. Thanks to the members of other groups in our university, and thank you for the time of online activities that we discussed and made progress together. We are especially grateful to Mr. Muthu Kumar Chandrasekaran for his patient guidance, forgiving and helping us when we first participated in the event. We thank each other, we have completed this project together, and it still has a lot to improve. We look forward to working together in the future. We are the best partner.

References

Tahamtan, Iman and Bornmann, Lutz. 2019. *What do citation counts measure? An updated review of studies on citations in scientific documents published between 2006 and 2018*. Scientometrics.

Qi Zhang, Xiangwen Liao and Zhaochun Ren. 2019. *Information Retrieval: 25th China Conference, CCIR 2019, Fuzhou, China, September 20–22, 2019, Proceedings*.

George A. Miller. 1995. *WordNet: A Lexical Database for English.* Communications of the ACM Vol. 38, No. 11: 39-41.

Nitin Hardeniya, Jacob Perkins, Deepti Chopra, Nisheeth Joshi, Iti Mathur. 2016. *Natural Language Processing: Python and NLTK*.

Rada Mihalcea and Courtney Corley. 2006. *Corpus-based and Knowledge-based Measures of Text Semantic Similarity*.

Philip Stone. 2006. *General Inquirer Categories*. The Gallup Organization.

Corinna Cortes and Vladimir Vapnik. 1995. *Support-vector networks. Machine Learning,* 20(3):273–297, 1995.

Charu C. Aggarwal. 2014. *Data Classification: Algorithms and Applications.* Chapman & Hall/CRC Data Mining and Knowledge Discovery, Band 35

Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.