CL-SciSumm Shared Task - Team Magma

Héctor Martínez Alonso, Raheleh Makki, and Jia Gu
Hector.MartinezAlonso, Raheleh.MakkiNiri, Jia.Gu@thomsonreuters.com

Thomson Reuters Labs, Toronto ON, M5J 0A1 Canada

Abstract. Finding the cited text spans of a scientific article based on the citation text is a challenging task. In this paper, we present our novel system to identify cited sentence(s) and their residential sections in a reference paper, given a citing text. We define this task as a binary classification problem. We use domain-specific features obtained from ACL terminology. The predictions of the system are generated by a logistic regression classifier, with additional predictions from an Adaboost-decision tree added if the logistic regression predictions do not show sufficient diversity according to a threshold.

Keywords: Automatic summarization · scientific publications · terminology · lexical knowledge bases · logistic regression · citation

1 Introduction

Scientific researchers should have comprehensive knowledge of previous work and recent advancements in their field of interest. Considering the rapid growth of the number of scientific publications, it is a time-consuming and challenging task. Hence, automatic summarization has attracted many NLP researchers in recent years. One of the recent approaches is based on first finding all cited text spans that cite a paper and then creating a summary from those sentences [2,1,3]. The main subtask of this approach is to identify cited text spans given the citation sentences (citances).

The 4th Computational Linguistics (CL) Scientific Document Summarization Shared Task (CL-SciSumm 2018) is part of the BIRNDL workshop at the annual ACM SIGIR Conference and focuses on scientific document summarization in the CL domain. CL-SciSumm 2018 includes three subtasks of 1A) identifying cited sentence(s) for a given citance, 1B) determining in which facet (e.g. *Method* or *Implication*) the reference paper is being cited, and 2) generating a summary. This report presents our proposed approach for tasks 1A and 1B. We have used Scikit-Learn and NLTL for our Python implementation.[1]

2 Formulation for Task 1A

We treat Task 1A as a classification problem, namely as learning a function $f(c, r)$ that, given a citation offset c (a text span from the citing article), and

[1] Available at `https://github.com/hectormartinez/scisumm_tr_magma`

a reference offset r (a text span from the reference article), determines whether the citation sentence(s) cites the reference sentence(s).

We create the training dataset using the annotation files provided for the shared task. Every reference article has an annotation file containing all citances to it, and their cited text. Each citance c to reference article R can be paired with every sentence (cited text) in R, i.e. $\forall r \in R$. If the pair (c, r) exists in the corresponding annotation file, its label is positive; otherwise, its label is negative. If citance c in the annotation file is referring to multiple sentences $r_i..r_j$ in R, we consider all possible pairs between c and $r_i..r_j$ as positive instances. The resulting dataset is very skewed, as most reference sentences are not the ones that will end up being cited. Indeed, out of 180,867 training instances, only 753 of them are positive examples (0.4%).

For the test set, we follow the same steps to create the instances, namely pairing each citance c with every sentence r in the reference article. However, these labels are unknown and should be predicted by the trained model.

2.1 Features

In order to characterize our (c,r) pairs we use the following features. Let W be the set of words of a certain document that appear at least 20 times in the training data and are not stop words, we define W_r and W_c for citation and reference respectively.

Bag of words: We build a bag of words for W_r , and another for W_c, and we calculate the size of their intersection.

Brown clusters: We construct two feature spaces, for W_r and W_c respectively, and each one is a bag-of-words style space formed by the Brown clusters of the words in its set. We use the ACL Brown clusters distributed with [5].

Embeddings: We calculate the average embedding vector of W_r and of W_c, which yields two 100-dimensional feature spaces, and an additional numeric feature with the cosine of the vectors for W_r and W_c. We use the ACL corpus word embeddings distributed with [5].

Sentence scope and position in document: We calculate numeric features to give account for the number of year dates (e.g. 1997) in W_r, the number of capitalized words (potentially names) in W_r, the lengths of c and r, and the number of sentences in their respective offsets. We also calculate the relative position of r in its document, i.e. the index of r divided by the number of sentences in the document, as well as the relative position of r in it section. Finally, we add a small bag of words with the words in the section name for r, following the intuition that an abstract is less often cited than a methods section.

Terminology: We use the ACL terminological base provided in [5] to obtain features. The terminological base contains terms of arbitrary length in the domain of Computational Linguistics and Natural Language Processing. Each term is provided with a predicted label of a small set like *technology* or *linguistics*. We calculate the size of term overlap between c and r for n-grams of 4 or less, and the amount of terms in r. We construct three specialized bag-of-words style feature sets for the terms in c, the terms in r, and the ones they have in common.

Furthermore, we add two binary features to determine whether r contains no terms, and whether there are no terms in common between c and r.

WordNet: We use features from WordNet [4], chiefly, we replace all the words in W_r with their WordNet supersense and build a bag of words. Supersenses, also called lexnames or first beginners, are coarse semantic tags like *noun.person* or *verb.cognition*.

2.2 Model

Many of the features we use are bag-of-words based, and are prone to sparsity and overfitting in a skewed distribution. In addition to only considering frequent terms (appearing more than 20 times), all the features that are not present in the test set are removed from the training data. We perform five-fold cross-validation without shuffling the results to best simulate the effect of out-of-vocabulary ratio on new data.

We have experimented with different simple classifiers chiefly decision trees and logistic regression, and different ensemble methods derived thereof. Our exploration of SVM with a polynomial or RBF kernel did not outperform simpler classifiers or ensembles. Table 1 shows our two most competitive classifiers. The system marked in bold, LogregL2C10, is responsible for most of the submitted predictions, and Adaboost20 provides auxiliary predictions (cf. Section 2.4).

Table 1. Cross-validation results for candidate systems. Units given in percentage. Average is prevalence-weighted macro-average across classes.

Model	Class	Prec.	Recall	F1	Support
	−	99.61	100.0	99.80	180,114
	+	85.42	05.44	10.24	753
Adaboost20	Avg.	99.55	99.60	99.43	180,867
	−	99.64	99.87	99.75	180,114
	+	30.33	13.41	18.60	753
LogregL2C10	Avg.	99.35	99.51	99.42	180,867

2.3 Evaluation

We also report the results of applying the evaluation script provided by CL-SciSumm[2] for the two Logistic regression models (Table 2). As explained in Section 2, we report the sentence with the highest score as the cited text for each citance.

[2] https://github.com/WING-NUS/scisumm-corpus/blob/master/2018-evaluation-script/program/task1_eval.py

Table 2. Evaluation metrics for best classifiers. Units given in percentage. Across across the file folds.

Model	Micro-Avg			Macro-Avg		
	Prec.	Recall	F1	Prec.	Recall	F1
LogregL2C10	0.75	0.50	0.60	0.77	0.65	0.71
AdaBoost20	0.67	0.45	0.54	0.76	0.59	0.67

2.4 Post-processing of predictions

Our system normally chooses only 1 sentence in the reference for each citance. More specifically, we sort the sentences of the reference article by the probability scores predicted by the classifier, and select the sentence with the highest probability score as the cited text for a given citance *c*.

However, some citances might cite more than one reference sentence. We achieve multiple prediction by joining results of our two candidate classifiers: if the ratio between the number of citing sentences and the number of LogregL2C10-predicted reference sentences equals or exceeds 7:1, we add an additional reference sentence from the top of the Adaboost20 prediction scores. Adaboost20 is also a competitive classifier that has substantial diversity of classification criterion with regards to our main classifier, namely LogRecL2C10. Moreover, Adaboost20 has very high precision for the positive class and makes a good candidate for ensemble prediction. We have applied this extension of the prediction set in 4 out of the 20 reference documents that make up the test set.

3 Formulation for Task 1B

Task 1B requires labeling positive predictions to determine its *facet*, which can be *Aim*, *Hypothesis*, *Method*, *Result* or *Implication*. We have applied a heuristic labeling based on the section of the reference text. We construct a lookup from section names to their most frequent citation label, and apply it on test. If a section name is not present in the lookup, we back off to the *Method* label. The macro-averaged F1 score (in percentage) for LogregL2C10 and AdaBoost20 is 8.9 and 5.3 respectively.

4 Conclusion

We approached the problem of identifying the cited text for a given citance as a binary classification task. We derived our features using a combination of content-based features and similarity measures, and performed an extensive feature and model selection. The cross validation results on the training dataset show that Logistic regression with L2 regularization outperforms more complex ensemble or kernel-SVM models.

References

1. Jaidka, K., Chandrasekaran, M.K., Jain, D., Kan, M.Y.: 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)
2. Jaidka, K., Chandrasekaran, M.K., Rustagi, S., Kan, M.Y.: Insights from cl-scisumm 2016: the faceted scientific document summarization shared task. International Journal on Digital Libraries pp. 1–9 (2017)
3. Ma, S., Xu, J., Zhang, C.: Automatic identification of cited text spans: a multi-classifier approach over imbalanced dataset. Scientometrics pp. 1–28 (2018)
4. Miller, G.A.: WordNet: a lexical database for English. Communications of the ACM **38**(11), 39–41 (1995)
5. Schumann, A.K., Alonso, H.M.: Automatic annotation of semantic term types in the complete acl anthology reference corpus. LREC (2018)