University of Houston @ CL-SciSumm 2017: Positional language Models, Structural Correspondence Learning and Textual Entailment

Samaneh Karimi[1,2], Luis Moraes[2], Avisha Das[2], and Rakesh Verma[2]

[1] School of Electrical and Computer Engineering, University of Tehran, Iran
[2] Computer Science Department, University of Houston, TX 77204

Abstract. This paper introduces the methods employed by University of Houston team participating in the CL-SciSumm 2017 Shared Task at BIRNDL 2017 to identify reference spans in a reference document given sentences from citing papers. The following approaches were investigated: structural correspondence learning, positional language models, and textual entailment. In addition, we refined our methods from BIRNDL 2016. Furthermore, we analyzed the results of each method to find the best performing system.

1 Introduction

The CL-SciSumm 2017 shared task [11] focuses on the problem of automatic summarization of scientific papers in the Computational Linguistics domain. In this problem, inputs are a set of reference documents and sets of citing documents associated with each reference document. Moreover, in each citing document, sentences which refer to the reference document (called citances) are marked. There are a couple of tasks defined in the shared task. Task 1a is, given a citance, to identify the span of reference text that best reflects what has been cited. Task 1b asks us to classify the cited span according to a predefined set of facets: hypothesis, aim, method, results, and implication. Finally, Task 2 is generating a structured summary.
Three main approaches are investigated for Task 1a: positional language models, structural correspondence learning, and textual entailment systems. The details of each method are explained in the following sections. Two methods are employed to address task 1b: a rule-based method which is basically a comparison-based method augmented by WordNet expansion and a classification method.

2 Related Works

Citations are considered an important source of information in many text mining areas [9]. For example, citations can be used in summarization to improve a summary [23]. It is thought that citations embody the community's perspective on the content of said paper [22].

In [26], the authors illustrate the importance of citations for summarization purposes. They made their summaries based on three sets of information including only the reference article; only the abstract; and, only citations. Finally they showed that citations produced the best results. In another study, Mohammad et al. [20] also showed that the information from citations is different from that which can be gleaned from just the abstract or reference article. However, it is cautioned that citations often focus on very specific aspects of a paper [8].

Properly tagging/marking the actual citation has also attracted a great deal of attention to this area of reserach. Powley and Dale [25] give insight into recognizing text that is a citation. Siddharthan and Teufel also introduce a new concept called "scientific attribution" which can help in discourse classification. The importance of discourse classification is further developed in [1]; in this paper, they showed the importance of discourse facet identification for producing good summaries.

In terms of what has been attempted at CL-SciSumm in past years, the methods are diverse. Aggarwal and Sharma [2] use bag-of-words bigrams, and compute scores to rank reference sentences based on their relevance to the citance using bigram overlap counts between citance and reference sentences using some heuristics. In [12], researchers generate three combinations of an unsupervised graph-based sentence ranking approach with a supervised classification approach. Cao et al. [5] model Task 1a as a ranking problem and apply SVM Rank for this purpose. In [16], the citance is treated as a query over the sentences of the reference document; the authors then used learning-to-rank algorithms (RankBoost, RankNet, AdaRank, and Coordinate Ascent) with lexical and topic features, in addition to TextRank scores, for ranking sentences. Lei et al. [14] used SVMs and rule-based methods with lexicon features and similarities (IDF, Jaccard, and context similarity). In [24], authors propose a linear combination between a TFIDF model and a single layer neural network model. Saggion et al. [28] used supervised algorithms with feature vectors representing the citance and reference document sentences. Features include positional and rhetorical features, in addition to WordNet similarity measures.

3 Dataset

The dataset for CL-SciSumm 2017 [11] is divided into 30 training documents and 10 testing documents, each with multiple citances. In the rest of this section, some statistics about the raw dataset (with no preprocessing) are reported.

This dataset contains 148,669 words and 11,114 unique words among the reference documents. There are 6,700 reference sentences and their average length is 23 words. The average reference document length is 4955 words in this dataset. Furthermore, the average number of sentences in each reference document is approximately 223 sentences.

4 Task 1a Individual Methods

In this task, we are asked to identify the reference sentences referred to by a given citance. In general, we rank the sentences in the reference document according to some method, then return the top 3. This year, the following new methods were attempted by our team: Positional Language Models, Structural Correspondence Learning, and Textual Entailment techniques.

4.1 Positional Language Model Approach

Positional language model was proposed with the idea of employing proximity information in documents to retrieve better results in response to a query[17]. In task 1a, we consider each reference text a document and each citance a query. Using PLM approach, a separate language model is constructed for each position of the reference sentence and computes the score of the reference sentence based on the similarity between its positional language models and citances language model. The elements of PLM (Positional Language Model) are the propagated counts of all words within the reference sentence which are estimated using a density function. With this idea, the closer words to the position, the higher the weight of the word in the PLM. Therefore, the PLM of reference sentence d at position i is estimated as follows:

$$p(w|d,i) = \frac{c^{'}(w,i)}{\sum_{w^{'}\epsilon V} c^{'}(w^{'},i)}$$

where V is the vocabulary and $c^{'}(w,i)$ is the propagated count of word w at position i from all of its occurrences in the reference sentence. Finally, PLM of each position in the reference sentence is compared with the language model of citance using KL-divergence to obtain a position specific similarity score as follows:

$$S(q,d,i) = -\sum_{w \epsilon V} p(w|q) \log \frac{p(w|q)}{p(w|d,i)}$$

where $p(w|q)$ is the language model of the citance q, $p(w|d,i)$ is the positional language model of reference sentence d at position i and $S(q,d,i)$ is the similarity score between the position i in the reference document and the citance. These scores are used to find the final similarity score of reference sentence(as a document) in response to the citance(as a query). Thus, we can apply PLM approach as a retrieval process to find the most relevant reference sentences in response to each citance.

4.2 Structural Correspondence Learning Approach

SCL is a method of transfer learning that attempts to learn a joint representation for two different domains [4]. The reasoning behind using SCL in this task is that citances and the sentences to which they refer belong to different domains, yet correspond to each other. Thus, it seemed plausible that such a method would be beneficial.

Structural Correspondence Learning seeks to find a joint representation by focusing on *pivot features*, i.e. features that are frequent in occurrence in both domains. The key to SCL is to predict the occurrence of pivot features from the non-pivot features of an example. One can learn a machine learning model, such as an SVM, for this purpose. The next step is to reduce the dimensionality of these predictors; this forces some generalization. The joint representation consists of the predicted pivot features (non-pivot features are thrown away). For our purposes, these new feature vectors are used to calculate cosine similarity scores with the citance.

4.3 Textual Entailment Approach

The property of *textual entailment* between two pieces of text can be described as a directional relationship which can only be *True* when the information contained in one text fragment is directly or indirectly derived from the other text fragment. The derived text fragment is then said to be textually entailed by the other. In Textual Entailment[3], the entailing fragment is termed the *text* and the possibly entailed fragment is the *hypothesis*. For example, the following pair of text fragments demonstrate entailment:

Text:	`The cat ate the rat.`
Hypo.:	`The cat is not hungry.`

The task of deriving inference from pairs of text is called Recognizing Textual Entailment (RTE)[4]. The proposed approach uses textual entailment as a measure of extracting the reference sentences relevant to a given citance. We build textual pairs using the given citance (*text*) and the sentences extracted from the reference document (*hypothesis*). We use an RTE system **TIFMO** [7, 29] to measure textual entailment between a given pair of citance and reference text. TIFMO uses Dependency-based Compositional Semantics (DCS) [29] based trees to represent a text body. The system derives an inference for entailment prediction by considering logic based relations between *'abstract denotations'* or relational expressions generated from the queries in the DCS trees. A further improvement to the system was proposed in [7], where Generalized Quantifiers (GQs) present in text are taken into account to evaluate lexical and/or syntactical relations between pairs of sentences (text and hypothesis) to predict the presence of entailment between them and also the type of entailment. We have used the TIFMO system proposed in [7] for our evaluation of citances and extraction of their relevant reference sentences.

[3] `https://aclweb.org/aclwiki/Textual_Entailment_Portal`

[4] `https://aclweb.org/aclwiki/Recognizing_Textual_Entailment`

4.4 Previous Methods

We present an overview of the methods that were previously employed on this task in [21].

TFIDF. In this method we rank sentences in the reference document according to the cosine similarity between each sentence and the citance. Both the sentences and the citance are represented as a TFIDF vector, i.e. a word vector where the weights are the TFIDF values calculated from the reference document.
Although TFIDF has been evaluated before in [21], this time we experiment with using more than just unigrams. We include variations that make use of bigrams and trigrams as well. Our naming convention for these systems includes the range of n-grams they use (for example, tfidf-1:3 uses unigrams, bigrams, and trigrams).

LDA. Latent Dirichlet Allocation is a topic modeling method [3] that models the interaction between topics and words as a statistical process. Topics within this model are drawn from a multinomial distribution. In turn, every topic has its own multinomial distribution for the words in the vocabulary. Thus, the model can capture the fact certain topics favor certain words.
For our task, we represent each sentence by the topic membership vector, which assigns to the sentence a probability of membership for each topic. These vectors are then ranked by cosine similarity, similar to TFIDF.

Word Embeddings. Word embeddings assign to each word a real-valued vector [18]. Through continuous iteration, the similarity between these vectors starts to approximate the similarity between the words they represent. Thus, since ‘dog’ and ‘pet’ are similar, their respective real-valued vectors will be similar as well.
Our task concerns the similarity between sentences, however. To generate sentence similarities from word similarities, we employ the Word Mover’s Distance [13].
In addition to embeddings learned through the ACL anthology, we tested the performance of embeddings that were pretrained on the Google News corpus [18].

4.5 Evaluation

The evaluation of our systems in task 1a uses the metrics: Precision@3, Recall@3 and F_1-score. In addition to the results of the PLM method, two well known information retrieval methods including KL-divergence and Okapi are employed to compare with the results of PLM. In all of the retrieval methods and PLM method, reference sentences are documents and citances are queries. Okapi is known as a ranking function which is based on the probabilistic retrieval framework. KL-divergence is a language modeling retrieval approach which compares language models of document with the query and ranks them based on their KL-divergence score. The results of all three methods for task 1a on training and test set 2017 are reported in Table 1. Runs with an asterisk (*) were submitted. As Table 1 shows, TFIDF is still a top performer. A few of the results are different from previous work due to the fact these results are obtained from all 30 training documents. For instance, in comparison to the results in [21], LDA and word embeddings show worse performance. It is surprising that SCL performs better than LDA. TIFMO does not perform as well as expected. Positional language model is performing better than KL-divergence and Okapi. However, none of these methods perform desirably well. One of the important reasons for this performance can be the difficulty of queries which are citances in our problem definition. Since citances may not include any of the reference text's words, it makes the retrieval process more difficult.

	Train			Test
Method	P@3	R@3	F_1	F_1
tfidf-1:1*	11.05%	21.20%	14.53%	6.84%
tfidf-1:2	11.39%	21.85%	14.97%	7.70%
tfidf-1:3*	11.05%	21.20%	14.53%	6.84%
tfidf-2:3	10.86%	16.57%	13.12%	7.13%
word2vec*	10.88%	20.88%	14.31%	9.12%
LDA	2.63%	5.05%	3.46%	1.99%
SCL	4.03%	6.02%	4.13%	2.28%
TIFMO	2.02%	3.88%	2.66%	1.99%
PLM	3.03%	5.81%	3.98%	0.84%
KL-div	2.63%	5.05%	3.46%	0.84%
Okapi	3.03%	5.81%	3.98%	1.13%

Table 1. Scores for individual systems on the 2017 dataset.

5 Task 1b

In Task 1b, for each cited text span, we pick the facet to which it belongs from a predefined set of facets. Two different approaches are employed in this task: A rule-based approach and a machine learning approach.

Rule-based Approach. The Rule-based approach consists of three consecutive steps. Each one is designed to find the correct facet through some comparisons, in case a match was not found in any of the previous steps. In the first step, citance words are compared with all five facet labels: Method, Implication, Result, Hypothesis and Aim. If none of the words in the citance match a facet label, then we move on to the second step. In the second step of the rule-based approach, an expanded form of the citance is compared with the facet labels. We expand the citance by adding all WordNet synsets [19] of each word found in the citance. In the third step, if no matched facet label is found in steps one and two, we expand the facet labels with their synsets and once again compare with the words in the citance.

Machine Learning Approach. In this approach, each citance is represented by a feature vector containing TFIDF values of its words and a classification model is learned using our training set. Then, the trained model is used to classify citances of the testing set. Machine learning methods used in this approach include Support Vector Machines (SVMs) [6], Random Forests [15], Decision Trees [27], MLP, and Adaboost [10].

5.1 Evaluation

As explained in section 5, we have employed two different approaches in Task 1b: a rule-based approach and a machine learning approach. The rule-based approach has different variations: 1) Rule_based-V1: In this variation, all three sets of comparisons (comparing citance words with facet labels, comparing expanded form of citances with facet labels and comparing expanded form of facets with citance words) are done while non-relevant synsets of all facets are excluded. 2) Rule_based-V2: In the second variation, all three sets of comparisons are done while only non-relevant synsets of “Method” facet are excluded. 3) Rule_based-V3: In the third variation, only first and second comparisons are done. The results of first approach of Task 1b on training set 2017 and testing set 2017 are represented in Table 2. A ”Method_only” approach which assigns ”method” to all of the citances is also employed to be compared with the rule-based approach.

	Train			Test
Method	P	R	F_1	F_1
Rule_based-V1	34.34%	31.43%	32.82%	28.84%
Rule_based-V2	58.41%	53.46%	55.83%	68.33%
Rule_based-V3	67.50%	61.70%	64.50%	78.99%
Method_only	69.36%	63.48%	66.29%	95.29%

Table 2. Recall, Precision, and F_1 score of rule-based method variations.

As Table 2 shows, the third variation of the rule-based approach outperforms other variations on both training and test sets. It means that expansion of facet labels does not help in finding the correct facet label of citances. Furthermore, the higher performance of Rule_based-V2 over Rule_based-V1 shows that excluding non-relevant synsets of the “Method” facet has a positive impact on the final results of the method. It might be due to the fact that “Method” is the most frequent facet label in both the training and test set for 2017. The results of the Method-only approach also verify this fact.

Table 3 shows the results of Task 1b for machine learning methods on the training and test set. For classification experiments on the training set, the training set is split into two separate datasets: a subset of 20 documents is used as train data and the remaining 10 documents are used as test data. For the classification experiments on the test set, the whole training set is used for the learning phase.

	Train			Test
Method	P	R	F_1	F_1
SVM	66.7%	59.0%	62.7%	73.35%
Random Forest	61.6%	54.5%	57.8%	72.50%
Decision Tree	50.0%	53.4%	51.6%	56.89%
MLP	61.4%	54.5%	57.7%	65.83%
Adaboost	54.0%	54.1%	54.1%	61.72%
Rule_based-V1	47.82%	42.30%	44.89%	28.84%
Rule_based-V2	63.24%	55.94%	59.36%	68.33%
Rule_based-V3	68.37%	60.48%	64.19%	78.99%
Method_only	69.16%	61.18%	64.93%	95.29%

Table 3. Recall, Precision, and F_1 score of classification methods.

As Table 3 shows SVM outperforms other classification methods in Task 1b and the lowest results among classification methods belongs to Decision Tree. Furthermore, comparison between the results of Table 2 and Table 3 shows that third variation of rule-based approach is our best performing method in Task 1b among all rule-based and classification methods.

6 Task 1a Method Combinations

In this section, we attempt to improve on the performance of the methods found in Section 4.5 by combining them. We combine methods in three ways: 1) through a linear combination of the methods, 2) through the use of one method as a “filter” for another, and 3) through the use of learning-to-rank algorithms that are fed the scores of our individual methods.

Linear Combination. A linear combination between two methods that tries to divide the importance given to the scores of two systems. An optimal tradeoff is calculated which normally generates better rankings than either system independently. For more details see [21].

$$\lambda \cdot sys1 + (1 - \lambda) \cdot sys2 \tag{1}$$

Filtering. The scores for one system are used to select the top N sentences from the reference document. These N sentences are then reranked according to another system. For $N = 3$ there will be no difference from the system that filters since we always return the top 3. However, as N increases, the rankings start to diverge.

Learning-to-Rank. We used a library of learning-to-rank algorithms, RankLib[5], to combine the scores generated by the other methods. We construct a modified dataset for use with RankLib. For each citance, we construct three different queries by subsampling the irrelevant sentences in the reference document. Therefore, each query consists of all relevant sentences (chosen by the annotator) and 10 irrelevant sentences chosen at random. This helps emphasize learning the ranking of the relevant sentences.

The scores of the following systems were used in conjunction: tfidf-1:1, tfidf-1:2, tfidf-1:3, tfidf-2:3, word2vec (ACL), word2vec (pretrained GoogleNews), SCL. These systems were chosen in an ad-hoc manner to provide a diverse set of competing rankings. Even though some of these systems underperform in general, they can occasionally provide better rankings for specific citances. No attempt was made to tune the hyperparameters for the algorithms.

Since learning-to-rank methods had a considerable jump in performance (as can be seen in Table 4), we had to test whether overfitting was occuring. We chose LambdaMART since it is similar to MART and obtained the second highest score when fed the whole training set. We sorted the training set documents by the number of annotated citances; every third document became part of a validation set. The performance gains measured were much more modest in this scenario.

6.1 Evaluation

The results obtained by combining individual systems are found in Table 4. Runs with an asterisk (*) were submitted. LambdaMART was chosen as the representative for the learning-to-rank algorithms and, thus, is the only algorithm with test set results.

[5] `https://sourceforge.net/p/lemur/wiki/RankLib/`

		Train		Test
Method	P@3	R@3	F_1	F_1
Linear Comb.*	11.79%	22.65%	15.51%	7.13%
Filtering*[6]	11.85%	22.76%	15.58%	7.41%
LambdaMART	21.71%	41.65%	28.55%	6.84%
Val. LambdaMART	13.08%	25.13%	17.21%	6.27%
MART	23.27%	44.66%	30.59%	–
Random Forest	16.01%	30.74%	21.05%	–
RankBoost	11.74%	22.54%	15.44%	–
ListNet	11.69%	22.43%	15.37%	–
Coord. Ascent	11.35%	21.79%	14.92%	–
RankNet	11.18%	21.46%	14.70%	–
Linear Regres.	11.07%	21.25%	14.56%	–
LambdaRank	0.00%	0.00%	0.00%	–

Table 4. Scores for combinations on the 2017 dataset.

7 Discussion

The results on the training set indicate semantic methods by themselves do not perform well, yet the test set's results directly contradict that claim: although TFIDF is the clear winner in the training set, the best method on the test set made use solely of word embeddings.

Task 1b also raises questions since the skewed facet distribution of the test set exacerbates the effectiveness of a simple baseline such as always choosing "Method". Regardless, we can choose better features for the classifiers that would permit us to reach a comparable level of performance.

In regard to the combination methods, there were less surprising results but still many unanswered questions. Our experiments with learning-to-rank methods hint at overfitting but the test set provided no evidence it was occurring. The linear combination between two systems, explored in [21], had similar performance to filtering. The filtering method between two systems was slightly more robust.

8 Future Work

We would like to investigate why there was such a drastic difference between the performance measured in the training and test sets. A comprehensive study can be done to contrast the characteristics of the training and test sets from a linguistic and statistical point-of-view. The differences between training and test set may reveal what type of citances benefit most from semantic information. In a similar vein, we would like to find out why there is a considerable difference between the performance of TFIDF and Okapi although they have similar formulations. Finally, we have not exhausted our exploration of textual entailmente and would like to investigate newer methods that have been developed.

[6] Submitted run was generated erroneously, which led to an F_1 score of 1.4%.

9 Acknowledgements

We would like to thank the NSF for Grants CNS 1319212, DGE 1433817 and DUE 1241772.

References

1. Amjad Abu-Jbara and Dragomir Radev. Coherent citation-based summarization of scientific papers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 500–509. Association for Computational Linguistics, 2011.
2. Peeyush Aggarwal and Richa Sharma. Lexical and syntactic cues to identify reference scope of citance. In *BIRNDL@ JCDL*, pages 103–112, 2016.
3. David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
4. John Blitzer, Ryan T. McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *EMNLP 2007, Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, 22-23 July 2006, Sydney, Australia*, pages 120–128, 2006.
5. Ziqiang Cao, Wenjie Li, and Dapeng Wu. Polyu at cl-scisumm 2016. In *BIRNDL@ JCDL*, pages 132–138, 2016.
6. Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
7. Yubing Dong, Ran Tian, and Yusuke Miyao. Encoding generalized quantifiers in dependency-based compositional semantics. 2014.
8. Aaron Elkiss, Siwei Shen, Anthony Fader, Güneş Erkan, David States, and Dragomir Radev. Blind men and elephants: What do citation summaries tell us about a research article? *Journal of the American Society for Information Science and Technology*, 59(1):51–62, 2008.
9. Aaron Elkiss, Siwei Shen, Anthony Fader, Gne Erkan, David States, and Dragomir Radev. Blind men and elephants: What do citation summaries tell us about a research article? *Journal of the American Society for Information Science and Technology*, 59(1):51–62, 2008.
10. Yoav Freund and Robert E. Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.
11. Kokil Jaidka, Muthu Kumar Chandrasekaran, Devanshu Jain, and Min-Yen Kan. Overview of the cl-scisumm 2017 shared task. In *Proceedings of Joint Workshop on Bibliometric-enhanced Information Retrieval and NLP for Digital Libraries (BIRNDL 2017), Tokyo, Japan, CEUR*, 2017.
12. Stefan Klampfl, Andi Rexha, and Roman Kern. Identifying referenced text in scientific publications by summarisation and classification techniques. In *BIRNDL@ JCDL*, pages 122–131, 2016.

13. Matt J Kusner, Yu Sun, Nicholas I Kolkin, Kilian Q Weinberger, et al. From word embeddings to document distances. In *ICML*, volume 15, pages 957–966, 2015.
14. Lei Li, Liyuan Mao, Yazhao Zhang, Junqi Chi, Taiwen Huang, Xiaoyue Cong, and Heng Peng. Cist system for cl-scisumm 2016 shared task. In *BIRNDL@ JCDL*, pages 156–167, 2016.
15. Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.
16. Kun Lu, Jin Mao, Gang Li, and Jian Xu. Recognizing reference spans and classifying their discourse facets. In *BIRNDL@ JCDL*, pages 139–145, 2016.
17. Yuanhua Lv and ChengXiang Zhai. Positional language models for information retrieval. *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval SIGIR 09*, page 299, 2009.
18. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
19. George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.
20. Saif Mohammad, Bonnie Dorr, Melissa Egan, Ahmed Hassan, Pradeep Muthukrishan, Vahed Qazvinian, Dragomir Radev, and David Zajic. Using citations to generate surveys of scientific paradigms. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 584–592. Association for Computational Linguistics, 2009.
21. Luis Moraes, Shahryar Baki, Rakesh Verma, and Daniel Lee. Identifying reference spans: topic modeling and word embeddings help ir. *International Journal on Digital Libraries*, 2017.
22. Preslav I Nakov, Ariel S Schwartz, and Marti Hearst. Citances: Citation sentences for semantic analysis of bioscience text. In *Proceedings of the SIGIR*, volume 4, pages 81–88, 2004.
23. Hidetsugu Nanba, Noriko Kando, and Manabu Okumura. Classification of research papers using citation links and citation types: Towards automatic review article generation. *Advances in Classification Research Online*, 11(1):117–134, 2000.
24. Tadashi Nomoto. Neal: A neurally enhanced approach to linking citation and reference. In *BIRNDL@ JCDL*, pages 168–174, 2016.
25. Brett Powley and Robert Dale. Evidence-based information extraction for high accuracy citation and author name identification. In *Large Scale Semantic Access to Content (Text, Image, Video, and Sound)*, pages 618–632. Le Centre de Hautes Etudes Internationales D'Informatique Documentaire, 2007.
26. Vahed Qazvinian, Dragomir R Radev, Saif Mohammad, Bonnie J Dorr, David M Zajic, Michael Whidby, and Taesun Moon. Generating extractive summaries of scientific paradigms. *J. Artif. Intell. Res.(JAIR)*, 46:165–201, 2013.
27. J. R. Quinlan. Induction of decision trees. *MACH. LEARN*, 1:81–106, 1986.

28. Horacio Saggion, Ahmed AbuRaed, and Francesco Ronzano. Trainable citation-enhanced summarization of scientific articles. In *Cabanac G, Chandrasekaran MK, Frommholz I, Jaidka K, Kan M, Mayr P, Wolfram D, editors. Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL); 2016 June 23; Newark, United States. CEUR Workshop Proceedings:[Sl]; 2016. p. 175-86.* CEUR Workshop Proceedings, 2016.

29. Ran Tian, Yusuke Miyao, and Takuya Matsuzaki. Logical inference on dependency-based compositional semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 79–89, 2014.