Identifying Referenced Text in Scientific Publications by Summarisation and Classification Techniques

Stefan Klampfl, Andi Rexha, and Roman Kern

Know-Center GmbH
Inffeldgasse 13, 8010 Graz, Austria
{sklampfl,arexha,rkern}@know-center.at

Abstract. This report describes our contribution to the 2nd Computational Linguistics Scientific Document Summarization Shared Task (CL-SciSumm 2016), which asked to identify the relevant text span in a reference paper that corresponds to a citation in another document that cites this paper. We developed three different approaches based on summarisation and classification techniques. First, we applied a modified version of an unsupervised summarisation technique, TextSentenceRank, to the reference document, which incorporates the similarity of sentences to the citation on a textual level. Second, we employed classification to select from candidates previously extracted through the original TextSentenceRank algorithm. Third, we used unsupervised summarisation of the relevant sub-part of the document that was previously selected in a supervised manner.

Keywords: text summarisation, key sentence extraction, citation analysis

1 Introduction

Extractive summarisation of a textual document is the process of finding a representative subset of the document text that captures as much information about the original document as possible. A promising idea in the realm of scientific publications is to consider the set of sentences that cite a paper as a summary created by the research community. Here we describe our contribution to the 2nd Computational Linguistics Scientific Document Summarization Shared Task (CL-SciSumm 2016)[1] [6], which aims at exploring and encouraging novel techniques for scientific paper summarisation along this direction. This task takes place at the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2016)[2] [2] at the Joint Conference on Digital Libraries (JCDL '16) and is a follow-up on the

[1] http://wing.comp.nus.edu.sg/cl-scisumm2016/
[2] http://wing.comp.nus.edu.sg/birndl-jcdl2016/

CL Pilot Task that has been conducted as a part of the BiomedSumm Track at the Text Analysis Conference 2014 (TAC 2014)[3] [5].

The dataset provided for this year's task consists of a set of reference papers (RP), each of which is accompanied by a set of citing papers (CP). The goal is to identify the text span in the RP which corresponds to the citations in CP (Task 1A) as well as the discourse facet of the RP this text span belongs to (Task 1B). Human annotators have created the ground truth in the form of pairs of citation text and cited text on the granularity level of sentences.

For Task 1A, we implemented a number of approaches that employ both unsupervised and supervised techniques that differ in the way how information from the citing sentence in the CP is incorporated into the process. In total, we submitted three runs, corresponding to our three approaches for Task 1A: (i) *modified-tsr*, (ii) *tsr-sent-class*, and (iii) *sect-class-tsr*.

First, in a completely unsupervised setting, we applied a modified variant of our TextSentenceRank algorithm to the RP (*modified-tsr*). TextSentenceRank [9] is a graph based ranking algorithm, a refinement of the well-known TextRank algorithm, which is applied to text in order to extract key words and/or key sentences. For the task at hand, we investigated a specific weighting that takes into account the information provided by the CP.

As a second approach for Task 1A, we employed a supervised classification setting following an unsupervised preprocessing (*tsr-sent-class*). Through the original version of TextSentenceRank we pre-selected candidate sentences from the RP, independently from the CP, that are potential text spans for being cited. For a given citation in the CP, we then selected the corresponding candidate through supervised classification.

In our third option, we took a dual approach (*sect-class-tsr*). We first used supervised learning to identify the relevant sub-part (section) of the RP that corresponds to the citing sentence in the CP. Once the relevant section has been found, we used the original version of TextSentenceRank on this sub-part, independent from the CP, to identify referenced text spans.

For Task 1B, we used a similar classifier as for identification of the section. We used features derived from the citing sentence as well es from the extracted text span in the reference document to determine the discourse facet. This is applied in all three approaches to Task 1A.

In principle, TextSentenceRank is able to extract multiple candidate text spans scattered across the document, but since the task description required the extraction of consecutive referenced text, we decided to output a single sentence as the extracted reference span in all of our approaches.

This report is structured as follows. In sections 2 and 3 we explain our approaches for both Task 1A and Task 1B in detail. In section 4 we individually evaluate the classifiers that we used in our approaches and present our results for the overall task. In the end, in section 5 we conclude, discuss our findings, and give an outlook to potential future work.

[3] http://www.nist.gov/tac/2014/BiomedSumm/

2 Task 1A: Identification of the Referenced Text Spans

In this subsection we detail our three approaches for Task 1A, the identification of referenced text spans. We also introduce TextSentenceRank as a base method which is used in all three runs.

2.1 TextSentenceRank as a Method for Extracting Candidate Text Spans

This approach is inspired by graph based ranking algorithms, such as Google's PageRank [1], where vertices in a graph are ranked based on their importance given by the connectedness within the graph: the more likely a vertex is visited by random walk, the higher is its score in the ranking. The *TextSentenceRank* algorithm [9] is an application of such a graph based ranking method to natural language text, returning a list of relevant key terms and/or key sentences ordered by descending scores. It builds a graph where vertices correspond to sentences or tokens and connects them with edges weighted according to their similarity (textual similarity between sentences or textual distance between words). Furthermore, each sentence vertex is connected to the vertices corresponding to the tokens it contains[4].

The score $s(v_i)$ of a vertex v_i is given by

$$s(v_i) = (1-d) + d \sum_{j \in I(v_i)} \frac{w_{ij}}{\sum_{v_k \in O(v_j)} w_{jk}} s(v_j), \tag{1}$$

where d is a parameter accounting for latent transitions between non-adjacent vertices (set to 0.85 as in [1, 7]), w_{ij} is the edge weight between v_i and v_j, and $I(v)$ and $O(v)$ are the predecessors and successors of v, respectively. The scores can be obtained algorithmically by an iterative procedure, or alternatively by solving an eigenvalue problem on the weighted adjacency matrix.

The TextSentenceRank algorithm is an extension to the original TextRank algorithm [7], which could either compute key terms or key sentences. It has been shown in [9] that by computing both key terms and key sentences at the same time, the performance of key term extraction can be improved.

When used for key sentence extraction, TextSentenceRank extracts the "most relevant" sentences in terms of how they are connected to other sentences in the document via co-occurring words. Here we pursue our intuition that such relevant sentences are also more likely to be cited and use TextSentenceRank as a base algorithm for extracting candidates for referenced text.

2.2 Run 1: A Modified Version of TextSentenceRank

Run 1 (*modified-tsr*) follows a completely unsupervised setting, where we applied a modified variant of our TextSentenceRank algorithm to the RP. The idea here

[4] Here, we restrict the set of relevant tokens to adjectives, nouns, and proper nouns. This information is obtained through part-of-speech tagging.

was to use a specific weighting of the underlying graph that takes into account the information provided by the CP, in particular, the citing sentence. More precisely, the weight of an edge adjacent to a node corresponding to sentence S is modified by

$$w_{\text{new}} = w_{\text{old}} * [1 + \text{sim}(S, C)], \tag{2}$$

where $\text{sim}(S, C)$ is a similarity measure between sentence S of the RP and citing sentence C of the CP.

In our approach we used the Jaccard similarity [4] on the sets of tokens contained in the respective sentences. Different similarity measures are possible, e.g., a measure capturing the semantic similarity of words in the sentences, but were not applied in the scope of this task. From the resulting list of the most relevant sentences in the RP, we selected the one with the largest similarity to the citing sentence in the CP.

2.3 Run 2: TextSentenceRank and Sentence Classification

In Run 2 (*tsr-sent-class*) we employed a supervised classification setting following an unsupervised preprocessing. First, we pre-selected candidate sentences from the RP through the original version of TextSentenceRank. This selection is thus independent from the CP and consists of potential text spans for being cited. We then selected the corresponding candidate through supervised classification that takes into account the information from the citing sentence in the CP.

We used a Random Forest classifier [3], an ensemble method based on decision trees, with the following features:

- **Section features:** title and number of the section enclosing the candidate sentence,
- **Sentence position features:** relative positions of the candidate sentence within the RP and within the enclosing section,
- **Discriminative term features:** information about tokens shared between the candidate sentence in the RP and the citing sentence in the CP.

This classifier is a binary classifier that decides for each candidate sentence in the RP whether it is an actual referenced text span, based on information from the citing sentence. For training the classifier we used the information provided by the training set. Because of the unbalanced nature of positive and negative training examples we used TextSentenceRank also to pre-select the sentences for training.

2.4 Run 3: Section classification and TextSentenceRank

In Run 3 (*sect-class-tsr*), we took a dual approach to Run 2. Instead of applying an unsupervised preprocessing followed by a supervised classification, we first used supervised learning to identify the relevant sub-part of the RP that corresponds to the citing sentence in the CP. This sub-part can in principle be of any granularity, but as sections are annotated in the provided dataset, we chose the granularity of sections. Once the relevant section has been found, we used the original version of TextSentenceRank on this sub-part, independent from the CP, to identify referenced text spans.

Again, we used a Random Forest classifier, now with these features:

- **Section features:** title and number of the section enclosing the candidate sentence,
- **Tf-Idf features:** information about the frequency of tokens from the citing sentence within the section, normalized by the inverse frequency across all the sections of the RP. This feature is motivated by the standard Tf-Idf measure [8], applied to the sections of the RP. It emphasizes sections that exclusively share tokens with the citing sentence.

This classifier is a binary classifier that decides for each candidate section in the RP whether it is a section containing a referenced text span, based on information from the citing sentence. The cited text span is then selected through the original version of TextSentenceRank, applied to the sub-document spanning the selected section, independently from the CP.

3 Task 1B: Identification of the Discourse Facet

The discourse facet takes the following values in the training set: *Implication*, *Method*, *Aim*, *Results*, and *Hypothesis*. We used a Random Forest classifier with section features, sentence position features, and discriminative term features to distinguish between these classes. That is, we took into account information from both the citing sentence as well es from the extracted text span in the reference document to determine the discourse facet. The same model, which was previously trained on the training set, was applied in all three runs.

4 Evaluation

In our evaluation, we first determine the isolated performance of individual components that we used in our approaches. Then we present our results for the overall task. We performed these evaluations on both the provided development set and the training set, as the results on the test corpus have not yet been made available.

4.1 Performance of individual classifiers

In our contribution we used three different classifiers, which we evaluate separately in this section. For the full system runs we trained the classifier on the training set, and submitted the results produced by applying the trained classifiers on the test set. Here, we evaluate the classifiers on both the development set and on the training set using 10-fold cross validation.

In Run 2 of Task 1A (*tsr-sent-class*) we used a binary classifier that decides for each candidate sentence whether it serves as a referenced text span for a given

Table 1. Performance of the sentence classifier on the development set and on the training set, evaluated by 10-fold cross validation. Precision, recall, and F1 values are given with respect to the positive class. Accuracy is the amount of correctly classified instances. The numbers in brackets denote the total number of instances for the respective scenario.

	Precision	**Recall**	**F1**	**Accuracy**
development set (2496)	0.803	0.221	0.346	0.926
training set (1338)	0.843	0.291	0.432	0.916

Table 2. Performance of the section classifier on the development set and on the training set, evaluated by 10-fold cross validation. Precision, recall, and F1 values are given with respect to the positive class. Accuracy is the amount of correctly classified instances. The numbers in brackets denote the total number of instances for the respective scenario.

	Precision	**Recall**	**F1**	**Accuracy**
development set (1093)	0.284	0.275	0.279	0.821
training set (561)	0.500	0.487	0.494	0.861

citing sentence. Table 1 shows the cross-validation performance of this sentence classifier on both the development set and on the training set. Since it is a binary classifier, we report here the precision, recall, and F1 values with respect to the positive class, i.e., whether the sentence is classified as a referenced text span. A reasonable precision is achieved, which means that there are relatively few false positives, however, at the expense of low recall, indicating that many true positives are missed. Even though we pre-filtered the negative instances with TextSentenceRank, the classification problem is still quite unbalanced: There are about 10 times as many negative as positive examples. This unbalance might induce a certain bias in the classifier; still, the accuracy, i.e., the fraction of correctly classified instances, across both classes is above 90% for both datasets.

In Run 3 of Task 1A (*sect-class-tsr*) we employed a binary classifier that decides for each section whether it contains a referenced text span corresponding to a given citing sentence. Table 2 shows the cross-validation performance of this section classifier on both the development set and on the training set. Again, we report here the precision, recall, and F1 values with respect to the positive class, i.e., whether the section is classified as containing a referenced text span. It can be seen that the performance is quite lower than for the sentence classifier, but the accuracy is still above 80%. Here the unbalance between positive and negative classes is given by the number of sections in the reference paper (on average about 5 to 6 in the training set).

In Task 1B, for the identification of the discourse facet, we used a multi-label classifier to categorise a referenced text span into one of the following classes:

Table 3. Performance of the discourse facet classifier on the development set and on the training set, evaluated by 10-fold cross validation. Precision, recall, and F1 values are given as micro-averages over the five classes. Accuracy is the amount of correctly classified instances. The numbers in brackets denote the total number of instances for the respective scenario.

	Precision	**Recall**	**F1**	**Accuracy**
development set (584)	0.602	0.707	0.628	0.707
training set (332)	0.698	0.696	0.666	0.696

Table 4. Confusion matrix of the discourse facet classifier on the training set, as well as precision and recall for each label, evaluated by 10-fold cross validation. Column headers denote classifier output, row headers denote true labels created by the human annotators.

	Implication	*Method*	*Aim*	*Results*	*Hypothesis*
Implication	13	14	5	0	0
Method	5	168	7	0	0
Aim	6	53	33	0	0
Results	0	7	0	17	0
Hypothesis	0	4	0	0	0
Precision	0.542	0.683	0.733	1.000	0.000
Recall	0.406	0.933	0.359	0.708	0.000

Implication, *Method*, *Aim*, *Results*, and *Hypothesis*. Table 3 shows the cross-validation performance of this discourse facet classifier on both the development set and on the training set. Precision and recall are given as micro-averages over all labels and lie between 60% and 70%. Classification accuracy is around 70% for both datasets. Table 4 shows the confusion matrix obtained by the classifier on the training set. *Method* is by far the most occurring label in the datasets, and the classifier might have a certain bias of generating this label, but also the quality of retrieving the label *Results* is reasonable. *Hypothesis* is the rarest label and the one with the lowest accuracy.

4.2 Overall task performance

We evaluated all of our three approaches for Task 1A on both the development set and the training set. We compared the extracted reference spans in our system output with the corresponding reference spans provided by the human annotators in terms of overlap and distance. In Table 5 we show for each of the three runs and for each topic in the development set the number of citances for which the extracted reference span lies within 10 sentences of the true reference span and for which both reference spans actually overlap. Table 6 shows the same information for the training set.

Table 5. Overall task performance on the development set. For each of the three runs and for each topic in the development set we show the number of citances for which the extracted reference span lies within 10 sentences of the true reference span. Numbers in brackets, if available, count citances where the extracted reference span overlaps the true reference span created by human annotators.

Topic	**Run 1**	**Run 2**	**Run 3**
W04-0213	2	1	1
P06-2124	10	0	1
E09-2008	6 (1)	0	3 (1)
W08-2222	2	4	2
C10-1045	5	0	0
W95-0104	2	0	1
C08-1098	1	1	0
D10-1083	1 (1)	0	0
N04-1038	3	0	2
C02-1025	6	3	1
Overall	38 (2)	9	11 (1)

It can be seen that only in few cases a referenced text span is extracted that overlaps the text segment identified by the human annotator. If we allow a certain neighborhood around the true spans, considerably more matches are found. Interestingly, Run 1, the modified TextSentenceRank, achieves the best results, followed by Run 3, the variant with section classification, which slightly outperforms Run 2, the version with sentence classification.

The finding that the modified TextSentenceRank works best suggests that considering the document as a whole might be beneficial for extracting relevant key sentences. The low performance of the classification approaches might be due to a lack of representative features that are relevant for the task at hand. In Run 2 it is possible that the set of sentences provided by the original TextSentenceRank algorithm is already too limited before the sentence classifier can select suitable text spans. The dual approach of Run 3, first selecting a sub-part and then applying TextSentenceRank, works slightly better.

These results also demonstrate the difficulty of this task. It is worth mentioning that all our approaches are solely based on statistics of words and sentences in both the reference document and the citing sentence as well as in their comparison. Currently, we do not incorporate any semantic information, but in principle our approaches can be easily adapted through additional features for classification and a different weighting strategy for TextSentenceRank.

5 Discussion

In this report we have described our contribution to the 2nd Computational Linguistics Scientific Document Summarization Shared Task (CL-SciSumm 2016),

Table 6. Overall task performance on the training set. For each of the three runs and for each topic in the training set we show the number of citances for which the extracted reference span lies within 10 sentences of the true reference span. Numbers in brackets, if available, count citances where the extracted reference span overlaps the true reference span created by human annotators.

Topic	**Run 1**	**Run 2**	**Run 3**
X96-1048	1 (1)	1	0
J00-3003	0	0	0
N01-1011	3 (2)	0	1
E03-1020	5 (1)	2	1 (1)
H89-2014	2	0	0
J98-2005	4	0	1 (1)
P98-1081	4 (1)	0	0
C90-2039	5 (1)	0	0
C94-2154	3 (2)	2	1
H05-1115	0	0	2
Overall	27 (8)	5	6 (2)

which asked participants to identify the relevant text span in a reference paper that corresponds to a citation in another document that cites this paper. We developed three different approaches based on summarisation and classification techniques. They employ both unsupervised and supervised techniques that differ in the way how information from the citing sentence is incorporated into the process. First, we applied a modified version of an unsupervised summarisation technique, TextSentenceRank, to the reference document, which incorporates the similarity of sentences to the citation on a textual level. Second, we employed classification to select from candidates previously extracted through the original TextSentenceRank algorithm. Third, we used unsupervised summarisation of the relevant sub-part of the document that was previously selected in a supervised manner.

We evaluated both the individual classifiers used in our approaches as well as the performance in the overall task. We believe that the performance of our systems could be improved by incorporating different similarity measures, e.g., measures capturing the semantic similarity of citing and cited sentences, not only in the modified weighting of TextSentenceRank, but also into the set of features used for classification. Furthermore, the relative strength of the influence of the citing sentence can be optimised. For example, in the modified TextSentenceRank algorithm there could be a trade-off parameter in equation 2 that weights the relative influences of w_{old} and $\text{sim}(S, C)$. Finally, the inclusion of multiple, non-consecutive sentences in the output would likely include candidate text spans of better quality.

Another aspect that likely influences our system is the fact that the text of the given documents was extracted with OCR methods. These methods some-times yield noisy and erroneous words, and many of our methods rely on the statistics of terms within and across documents. In our experience, in some cases TextSentenceRank seems to prefer sentences containing such noisy or invalid tokens.

As a contribution to the CL-SciSumm 2016 task our work aimed at facilitating the summarisation of scientific publications. In addition, we hope that our contribution will also provide further insight into the scientific writing habits of researchers, both in terms of how they structure their papers and how they reference the work of others.

Acknowledgements

The Know-Center is funded within the Austrian COMET Program – Competence Centers for Excellent Technologies – under the auspices of the Austrian Federal Ministry of Transport, Innovation and Technology, the Austrian Federal Ministry of Economy, Family and Youth and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

References

1. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Computer Networks 56(18), 3825–3833 (2012)
2. Cabanac, G., Chandrasekaran, M.K., Frommholz, I., Jaidka, K., Kan, M.Y., Mayr, P., Wolfram, D.: Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2016)
3. Ho, T.K.: Random decision forests. In: Proceedings of the 3rd International Conference on Document Analysis and Recognition. pp. 278–282. Montreal, QC (1995)
4. Jaccard, P.: The distribution of the flora in the alpine zone. New Phytologist 11, 37–50 (1912)
5. Jaidka, K., Chandrasekaran, M.K., Elizalde, B.F., Jha, R., Jones, C., Kan, M.Y., Khanna, A., Molla-Aliod, D., Radev, D.R., Ronzano, F., et al.: The computational linguistics summarization pilot task. In: Proceedings of TAC. Gaithersburg, USA (2014)
6. Jaidka, K., Chandrasekaran, M.K., Rustagi, S., Kan, M.Y.: Overview of the 2nd Computational Linguistics Scientific Document Summarization Shared Task (CL-SciSumm 2016). In: Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2016). Newark, New Jersey, USA (2016), to appear
7. Mihalcea, R., Tarau, P.: Textrank: Bringing order into texts. In: Conference on Empirical Methods in Natural Language Processing. Barcelona, Spain (2004)
8. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc., New York, NY, USA (1986)
9. Seifert, C., Ulbrich, E., Kern, R., Granitzer, M.: Text representation for efficient document annotation. Journal of Universal Computer Science 19(3), 383–405 (2013)