

Trainable Citation-enhanced Summarization of Scientific Articles

Horacio Saggion, Ahmed AbuRa'ed, Francesco Ronzano

TALN - DTIC
Universitat Pompeu Fabra
Barcelona, Spain

horacio.saggion@upf.edu, ahmed.aburaed@upf.edu, francesco.ronzano@upf.edu

Abstract. In order to cope with the growing number of relevant scientific publications to consider at a given time, automatic text summarization is a useful technique. However, summarizing scientific papers poses important challenges for the natural language processing community. In recent years a number of evaluation challenges have been proposed to address the problem of summarizing a scientific paper taking advantage of its citation network (i.e., the papers that cite the given paper). Here, we present our trainable technology to address a number of challenges in the context of the 2nd Computational Linguistics Scientific Document Summarization Shared Task.

1 Introduction

During the last decade the amount of scientific information available on-line increased at an unprecedented rate with recent estimates reporting a new paper published every 20 seconds [17]. In this scenario of scientific information overload, researchers are overwhelmed by an enormous and continuously growing number of articles to consider in their research work: from the exploration of advances in specific topics, to peer reviewing, writing and evaluation. In order to cope with the growing number of relevant publications to consider at a given time, automatic text summarization is a useful technique [23]. However, generic text summarization techniques may not work well in specialized genres such as the scientific genre and domain specific techniques may be needed [25, 26]. Scientific publications are characterized by several structural, linguistic and semantic peculiarities. Articles include common structural elements (title, authors, abstract, sections, figures, tables, citations, bibliography) that often require specific text processing tools. Additionally, scientific documents have specific discourse structure [27, 12]. Another important aspect of scientific papers is their network of citations that identifies links among research works, making them also particularly interesting from the social viewpoint. Although citation counts had been used to assess some aspects of research output for a long time, citation semantics has started to be exploited in several context including opinion mining [28, 1] and scientific text summarization [18, 2].

Considering the urgent need for new, automated approaches to browse and aggregate scientific information, in recent years a number of natural language processing challenges have been proposed: the Biomedical Summarization Task (BioSumm2014) carried out in the context of the Text Analysis Conferences¹ provided a forum for researchers interested in exploring the summarization of clusters of documents where one of the documents is a *reference paper* and the rest of the documents in the cluster are *citing papers* which cite the reference paper. In particular, the BioSumm2014 evaluation released a dataset consisting of 20 collections of annotated papers (i.e., clusters), each one including a reference article and 10 citing articles. Similarly, a pilot task on summarization of Computational Linguistic papers was proposed in 2014 [8]. Unfortunately, none of the evaluation contests provided with official evaluation results.

In this paper, we report our efforts to develop a system to participate in the CL-SciSumm 2016 evaluation [9] which is a renewed effort to address the challenges proposed in 2014. In a nutshell, participants were given a set of clusters, each one composed of n documents where one is a reference paper (RP) and the $n-1$ remaining documents are referred to as citing papers (CPs) since they cite the reference paper. Participants have to develop automatic procedures to perform the following tasks:

- **Task 1A:** For each citance (i.e., a reference to the RP), identify the spans of text (cited text spans) in the RP that most accurately reflect the citance.
- **Task 1B:** For each cited text span, identify what **facet** of the paper it belongs to, from a predefined set of facets, namely: Aim, Hypothesis, Implication, Results or Method.
- **Task 2:** Finally, an optional task consists on generating a structured (of up to 250 words) summary of the RP from the cited text spans of the RP.

In the rest of this paper we first present related work on summarization of research articles to then explain how we have addressed the different summarization tasks.

2 Related work

Although research in summarization can be traced back to the 50s [14] and although a number of important discoveries have been produced in this area, automatic text summarization still faces many challenges given its inherent complexity. Scientific text summarization is of paramount importance and scientific texts were automatic summarization's first application domain [14, 5]. Several methods and techniques have been already reported in the literature to produce text summaries by automatic means [23]. Summarization of scientific documents has been addressed from different angles: in [26] summarization is treated as

¹ <http://www.nist.gov/tac/2014/BiomedSumm/>

a rhetorical classification task where each sentence in an input text is classified as belonging to specific rhetorical categories (background, objective, etc.). Although the approach is interesting from the point of view of document interpretation, it is not a proper summarization task since no summary of the input is produced. [25] addressed the summarization problem as one of information extraction and text generation: the idea behind the approach is that a number of important concepts and relations should be extracted from text in order to create a summary independently of the particular scientific domain of the text. In recent years new generations of scientific summarization approaches have emerged which take advantage of the citations that a research paper has in order to extract and summarize its main contributions [19]. Methods to improve the coherence of the generated citation summaries use sentence classification to decide what type of information a sentence is conveying [1].

3 Transforming the Source Documents into GATE Language Resources

CL-SciSumm 2016 Challenge organizers have provided training data structured in clusters of reference and citing papers together with manual annotations indicating for each citance to the reference paper, the facet of this citance and the text span(s) in the reference paper that best represent the citance. In order to properly analyze the provided training and testing documents, we transformed the provided clusters into GATE documents. Given the manual annotations provided in text files, we automatically annotated the training set by creating in the reference paper a *References Annotation* set which contains the annotations corresponding to the text spans being cited by each citing paper. On the other hand, an *Annotation set* for the citances in each of the citing papers was also created. The link between citing paper annotations and reference paper annotations is implemented through a unique identifier (a concatenation of citance number, reference paper, citing paper, and annotator).

Such annotations are helpful in order to retrieve the necessary information from the documents. In this way, in each citing paper we are able to identify for each sentence that belongs to a citance, which are the sentences of the corresponding reference paper that most accurately reflect the citance. Thanks to this information, we can build pairs of matching sentences (Citing Paper Sentence, Reference Paper Sentence) and associate to each pair the facet that each annotator considers the citation is referring to (see Task 1B).

An example of the representation can be seen in Figure 1 where it is shown a reference paper (on the left side of the Figure) annotated with information from the citing papers (on the right side of the Figure).

3.1 Text Processing

Each document was annotated using processing resources from the GATE system [15] and the SUMMA library [22]. Additionally, and in order to further

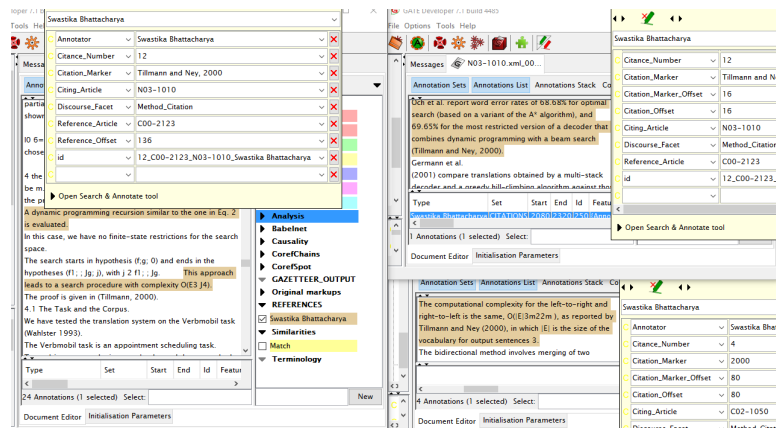


Fig. 1. GATE GUI representation of the annotations of a reference paper (left side) and two citing papers from the same cluster (right side).

enrich the documents, some components from the freely-available Dr Inventor library² (DRI Framework) were used [21]. The GATE system was used to tokenize, sentence split, part of speech tag, and lemmatize each document. The SUMMA library was used to produce normalized term vectors for each document (see Section 5). Although the Dr Inventor’s library produces very rich information, for the experiments we present here we rely only on its rhetorical sentence classification capabilities. The DRI Framework classify each sentence of a paper as belonging to a rhetorical category of scientific discourse among: Approach, Background, Challenge, Outcome and FutureWork. In particular, the framework computes for each sentence the probability the sentence has to belong to each rhetorical category. See [6] for details about the corpus used for training the classifier. For each sentence in the reference paper, we computed the cosine similarity between its sentence vector (see Section 5) and the vectors corresponding to the sentences citing the reference paper in the citing articles. These values were stored in the reference paper for further processing.

4 Method

In order to identify reference paper text spans for each citance (Task 1A), we modeled pairs of reference and citance sentences as a feature vector. Then, we used such pair representation to enable the training of distinct binary classification algorithms tailored to determine whether they are a match.

On the other hand we used the same representation of pairs of sentences for identifying to what facet of the reference paper a cited text span belongs to

² <http://backingdata.org/dri/library/>

(Task 1B): we classified each pair of sentences in one out of 5 predefined facets Aim, Hypothesis, Implication, Results or Method.

To this end, we relied on the Weka machine learning framework [29]. We evaluated the performance of six classification algorithms: SMO, Naive Bayes, J48, Lazy IBK, Decision table and Random Forest for both tasks. We performed 10-fold cross validation experiments with the training data in order to decide which algorithm to use during testing.

In the remainder of this Section, we describe the set of sentence pair features we used, and motivate their relevance with respect to the characterization of sentences similarity. When presenting the features, we group subsets of related features in the same subsection (Position features, Similarity features, etc.).

4.1 Position Features

We exploited the following set of position related features for both Task 1A (text spans for citance sentence) and 1B (the facet such text span belongs to):

- *Sentence position* (**sentence_position**): the normalized position of the sentence in the reference paper.
- *Sentence section position* (**sentence_section_position**): the normalized position of the sentence in the section of the reference paper.
- *Facet position* (**facet_aim, facet_hypothesis, facet_implication, facet_method** and **facet_result**): five features were generated to indicate which facet a cited text span belongs to. Binary values were calculated by analyzing the reference paper sentence’s section title and looking for any words which could indicate the feature facet: aim, hypothesis, implication, method or result. The value of the feature is 1 for section titles containing a word that indicate such facet and 0 otherwise.

4.2 WordNet Semantic Similarity Measures features

The following set of Semantic Similarity features were exploited for task 1A (text spans for citance sentence) only with the exception of the cosine similarity which was used for both task 1A and task 1B. We used WS4J (WordNet Similarity for Java) library which includes several semantic relatedness algorithms that rely on WordNet 3.0. Given a pair of sentences (reference and citance), we retrieve all the synsets associated to nouns and verbs in each one of them. Then, by considering all the pairs of synsets belonging to different sentences, we compute similarity values between citance sentence and reference sentence as follows³:

- *Path similarity* [7] (**path_similarity**): The shorter the path between two words/senses in WordNet, the more similar they are.

³ We calculated similarity values between each token in the citance sentence and each and every token in the reference sentence. Finally averaging all the similarities for the given sentence pair.

- *JCN similarity* [10] (**jiangconrath_similarity**): the conditional probability of encountering an instance of a child-synset given an instance of a parent synset.
- *LCH similarity* [11] (**lch_similarity**): the length of the shortest path between two synsets for their measure of similarity.
- *LESK similarity* [3] (**lesk_similarity**): Similarity of two concepts is defined as a function of the overlap between the corresponding definitions (i.e., their WordNet glosses).
- *LIN similarity* [13] (**lin_similarity**): The Similarity between A and B is measured by the ratio between the amount of information needed to state the commonality of A and B and the information needed to fully describe what A and B are.
- *RESNIK similarity* [20] (**resnik_similarity**): The probability of encountering an instance of concept c in a large corpus.
- *WUP similarity* [30] (**wup_similarity**): The depths of the two synsets in the WordNet taxonomies, along with the depth of the lowest common subsumer.
- *Cosine similarity* (**cosine_similarity**): The cosine similarity between the normalized vectors of the two sentences in the instance pair (this computation is different from the other similarity features).

4.3 Rhetorical Category Probability Features

We exploited the following set of rhetorical features for both task 1A (text spans for citance sentence) and 1B (the facet which the text span belongs to):

- *Rhetorical Category Probability* (**probability_approach**, **probability_background**, **probability_challenge**, **probability_future_work** and **probability_outcome**): five features were exploited to represent the probability of the reference text span to belong to such facet (from the Dr Inventor corpus and computed from the Dr Inventor library).

We also added both the reference sentence string and the citance sentence string to the set of features and then converted them to word vectors by using WEKA (i.e., bag-of-words).

4.4 Matching Citations to Reference Papers

The training data was prepared as follows: positive instances of the problem were the pairs of sentences from the citance which were matched with cited text spans from the references (according to information given in the gold annotations). Negative instances, instead, were pairs of sentences from citances to identified cited text spans which were not annotated as matches by the annotators (complementary information). As a consequence we casted the Task 1A as a binary classification problem where we decide for each pairs of citance sentence and reference paper sentence whether they match or not, or in other words whether the reference paper sentence reflects the reason of that specific

Class	Precision	Recall	F-Measure
Sent. Match	0.674	0.293	0.408
Sent. NoMatch	0.916	0.982	0.948
Averages	0.888	0.904	0.886

Table 1. J48 performance on testing data (10-fold cross validation) for the citance/reference matching problem (Task 1A). Last row of the table contains weighted average values.

citations. These procedure, which was decided upon to reduce the number of negative cases, produced 3,786 instances unevenly distributed (3,356 no matches vs 430 matches). After testing several algorithms from WEKA, we opted for the J48 implementation of decision threes. Ten fold cross-validation results are presented in Table 1.

4.5 Citation Purpose Identification

The training data was prepared as follows: similarly to the previous task, pairs of citing sentences and matched cited sentences (according to the gold annotations) were used to create instances. The facet of each instance was also given by the gold standard. This procedure produced just 432 instances with the following distribution: Aim (72), Implication (26), Result (76), Hypothesis (1), Method (257). After testing several algorithms from WEKA, we opted for the Support Vector Machines (SMO) implementation provided by the tool. We used polynomial Kernels and performed no parameter optimization due to time constraints. Ten fold cross-validation results are presented in Table 2.

Class	Precision	Recall	F-Measure
Aim	0.886	0.861	0.873
Implication	0.875	0.808	0.84
Results	0.971	0.895	0.932
Hypothesis	0.0	0.0	0.0
Method	0.929	0.969	0.949
Averages	0.924	0.926	0.924

Table 2. SMO performance on testing data (10-fold cross validation) for the facet identification problem (Task 1B). Last row of the table contains weighted average values.

5 Summarizing Scientific Articles

In order to summarize the reference paper by taking into account how it is mentioned in the citing papers, we combined information from the reference and

citing papers. We have implemented, using the resources of the freely available text summarization library SUMMA [22, 24], a series of sentence relevance features, all numeric, which are used to train a linear regression model following the methodology that was already used in [4].

In addition to rich set of features provided by the DRI Framework, document processing for summarization is carried out with SUMMA on reference and citing papers. More specifically, the following computations with the library are carried out to enable the summarization of scientific documents:

- Each token (i.e., lemma) is weighted by its term frequency* inverted document frequency, where inverted document values are computed from training data previously analysed (test documents in the CL-SciSumm 2016 dataset);
- For each sentence a vector of terms and normalized weights is created using the previously computed weights;
- For the title, a single vector of terms and normalized weights is also created (*title vector*);
- Using the normalized sentence term vectors in the whole document a centroid vector of terms is computed (*document centroid*);
- Using the normalized sentence term vectors of the abstracts a centroid vector of terms is computed (*abstract centroid*);
- All vectors corresponding to sentences citing the reference paper (from all citing papers) are used to create a centroid (*citances vector*).

The following is the set of sentence relevance features we have used for training a linear regression summarization system. Note that all text-based similarities we mention are the result of comparing two vectors using the cosine similarity function implemented in SUMMA. The reference paper features are as follows:

- *Sentence Abstract Similarity* (**abs_sim**): the similarity of a sentence to the author abstract;
- *Sentence Centroid Similarity* (**centroid_sim**): the similarity of a sentence to the document centroid (e.g., the average of all sentence vectors in the document);
- *First Sentence Similarity* (**firt_sim**): the similarity of a sentence to the title vector;
- *Position Score* (**position_score**): the SUMMA implementation of the position method where sentences at the beginning of the document have high scores and sentence at the end of the document have low scores;
- *Position in Section Score* (**in_sec**): an score representing the position of the sentence in the section of the document. Sentences in first section get higher scores, sentences in last section get low scores;
- *Sentence Position in Section Score* (**in_sec_sent**): a position method applied to sentences in each section of the document (sentence at the beginning of the section get higher scores and sentences at the end of the section get lower scores);
- *Normalised Cue-phrase Score* (**norm_cue**): we produce a normalized score for each sentence which is the total number of cue-words in the sentence

divided by the total number of cue-words in the document. We have relied on [26] formulaic expressions to implement our cue-phrase gazetteer lookup procedure;

- *TextRank Normalized Score* (**textrank_score**): the SUMMA implementation of the TextRank algorithm [16] but with a normalization procedure which yields values for sentences between 0 and 1.

The cluster-based features are as follows:

- *Citing Paper Maximum Similarity* (**cps_max**): each reference paper sentence vector is compared (using cosine) to each citance vector in each citing paper to obtain the maximum possible cosine similarity;
- *Citing Paper Average Similarity* (**cps_avg**): the average cosine similarity between a reference paper vector and all citance vectors in the cluster is produced;
- *Citing Paper Citances Similarity* (**cps_sim**): the similarity of the sentence vector to the centroid of the citance vectors.

The approach taken to score sentence is to produce a cumulative score of the weighted values of summarization features f_1, \dots, f_n using the following formula:

$$score(S) = \sum_{i=0}^n w_i * f_i \quad (1)$$

with S as the sentence to score, f_i as the value of feature i and w_i as the weight assigned to feature i . As we stated before, the weights of each features in the formula are learned from training data. We fit a linear regression model using the 10 testing documents from the provided annotated document for a total of 2,585 instances. The target numerical value to learn is computed from two sources (giving rise to two different systems): On the one hand, we compute the similarity of each reference paper sentence (i.e. vector) to the combined vectors of texts fragments identified as the annotators as cited text spans; on the other hand, we compute the similarity of each reference paper sentence (i.e. vector) to a vector of the community-based summary provided for training by the organizers. Table 3 shows the weights of the features learnt by the linear regression implementation from WEKA [29].

6 The Final System

The final system was assembled as follows. Given a cluster of documents with reference and citing papers, the following pipeline was applied for tasks 1A and 1B.

1. The documents were annotated with the citance information (no matched reference sentences were annotated);
2. All the document processing algorithms were applied to reference and citing papers as described in Section 3.1 and the features computed;

Feature	Citance Relevance	Community Relevance
abs_sim	0.0843	-0.0751
centroid_sim	0.7231	0.5795
cps_avg	0.0	0.3984
cps_max	0.1111	0.0
cps_sim	0.27	0.2806
first_sim	-0.0359	0.1801
in_sec	0.0	-0.0287
norm_cue	0.0921	0.1497
position_score	0.0483	0.0611
textrank_norm	-0.1622	-0.2251
Corr.	0.88	0.78

Table 3. Linear regression learnt weights for two conditions: relevance to cited text spans and relevance to a community-based summary. Last row indicates correlation coefficient of the model (in 10-fold cross-validation).

3. Instances were created using a citance sentence from each citing paper and each sentence from the reference paper;
4. The instances were sent to the matching classifier which returned a match/no match class and a confidence value;
5. The matched instances according to the previous steps were sent to the facet classifier to obtain the predicted citation facet.

Two runs were produced for tasks 1A and 1B. In one run, all matched sentences for a given citance were returned. In a second run, only top matches (with higher confidence) were returned. In order to produce the summaries for each cluster, summarization features were computed using the procedure described in Section 5, and SUMMA was exploited to score and extract top scored sentences based on formula (1). Two 250-word text summaries were produced per cluster using the models described in Section 5.

7 Outlook

In this paper, we have presented the techniques used to participate in the Computational Linguistics Summarization challenge. We have relied on competitive text processing and summarization tools to compute features to create rich document representations for dealing with the proposed tasks. Our approach is supervised combining evidence from several sources. Due to time limitations we could not carry out an exhaustive performance and feature analysis on test/development data, which we intend to carry out as future work. We look forward to know the official results of the evaluation so as to better understand the pros and cons of our approach.

Acknowledgements

This work is (partly) supported by the Spanish Ministry of Economy and Competitiveness under the Maria de Maeztu Units of Excellence Programme (MDM-

2015-0502), the TUNER project (TIN2015-65308-C5-5-R, MINECO/FEDER, UE) and the European Project Dr. Inventor (FP7-ICT-2013.8.1 - Grant no: 611383).

References

1. Abu-Jbara, A., Ezra, J., Radev, D.R.: Purpose and polarity of citation: Towards nlp-based bibliometrics. In: HLT-NAACL. pp. 596–606 (2013)
2. Abu-Jbara, A., Radev, D.: 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. pp. 500–509. HLT '11, Association for Computational Linguistics, Stroudsburg, PA, USA (2011)
3. Banerjee, S., Pedersen, T.: An adapted lesk algorithm for word sense disambiguation using wordnet. In: Computational linguistics and intelligent text processing, pp. 136–145. Springer (2002)
4. Brüggmann, S., Bouayad-Aghab, N., Burga, A., Carrascosa, S., Ciaramella, A., Ciaramella, M., Codina-Filba, J., Escorsa, E., Judea, A., Mille, S., Müller, A., Saggion, H., Ziering, P., Schütze, H., Wanner, L.: Towards content-oriented patent document processing: Intelligent patent analysis and summarization. *World Patent Information* 40, 30–42 (2015)
5. Edmundson, H.P.: New methods in automatic extracting. *J. ACM* 16(2), 264–285 (Apr 1969)
6. Fisas, B., Ronzano, F., Saggion, H.: A Multi-Layered Annotated Corpus of Scientific Papers. In: Calzolari, N., Choukri, K., Declerck, T., Grobelnik, M., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). European Language Resources Association (ELRA), Paris, France (may 2016)
7. Hirst, G., St-Onge, D.: Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet: An electronic lexical database* 305, 305–332 (1998)
8. 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., Saggion, H.: The computational linguistics summarization pilot task. In: Proceedings of TAC 2014 (2014)
9. 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: To appear in the Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2016) (2016)
10. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. arXiv preprint [cmp-lg/9709008](https://arxiv.org/abs/9709008) (1997)
11. Leacock, C., Chodorow, M.: Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database* 49(2), 265–283 (1998)
12. Liakata, M., Teufel, S., Siddharthan, A., Batchelor, C.R., et al.: Corpora for the conceptualisation and zoning of scientific papers. In: LREC (2010)
13. Lin, D.: An information-theoretic definition of similarity. In: ICML. vol. 98, pp. 296–304 (1998)
14. Luhn, H.P.: The automatic creation of literature abstracts. *IBM J. Res. Dev.* 2(2), 159–165 (Apr 1958)

15. Maynard, D., Tablan, V., Cunningham, H., Ursu, C., Saggion, H., Bontcheva, K., Wilks, Y.: Architectural Elements of Language Engineering Robustness. *Journal of Natural Language Engineering – Special Issue on Robust Methods in Analysis of Natural Language Data* 8(2/3), 257–274 (2002)
16. Mihalcea, R., Tarau, P.: TextRank: Bringing order into texts. In: *Proceedings of EMNLP-04 and the 2004 Conference on Empirical Methods in Natural Language Processing* (July 2004)
17. Munroe, R.: The rise of open access. *Science* 342(6154), 58–59 (2013), <https://www.sciencemag.org/content/342/6154/58.full>
18. Qazvinian, V., Radev, D.R.: Identifying non-explicit citing sentences for citation-based summarization. In: *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*. pp. 555–564 (2010)
19. Qazvinian, V., Radev, D.R.: Identifying non-explicit citing sentences for citation-based summarization. In: *Proceedings of the 48th annual meeting of the association for computational linguistics*. pp. 555–564. Association for Computational Linguistics (2010)
20. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. *arxiv preprint cmlg/9511007* (1995)
21. Ronzano, F., Saggion, H.: Dr. inventor framework: Extracting structured information from scientific publications. In: *Discovery Science - 18th International Conference, DS 2015, Banff, AB, Canada, October 4-6, 2015, Proceedings*. pp. 209–220 (2015)
22. Saggion, H.: SUMMA: A Robust and Adaptable Summarization Tool. *Traitement Automatique des Langues* 49(2), 103–125 (2008)
23. Saggion, H., Poibeau, T.: Automatic text summarization: Past, present and future. In: Poibeau, T., Saggion, H., Piskorski, J., Yangarber, R. (eds.) *Multi-source, Multilingual Information Extraction and Summarization*. Springer Verlag, Berlin (2013)
24. Saggion, H.: Creating summarization systems with SUMMA. In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), Reykjavik, Iceland, May 26-31, 2014*. pp. 4157–4163 (2014)
25. Saggion, H., Lapalme, G.: Generating indicative-informative summaries with sum. *Comput. Linguist.* 28(4), 497–526 (Dec 2002)
26. Teufel, S., Moens, M.: Summarizing scientific articles: Experiments with relevance and rhetorical status. *Comput. Linguist.* 28(4), 409–445 (Dec 2002)
27. Teufel, S., Siddharthan, A., Batchelor, C.: Towards discipline-independent argumentative zoning: evidence from chemistry and computational linguistics. In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*. pp. 1493–1502. Association for Computational Linguistics (2009)
28. Teufel, S., Siddharthan, A., Tidhar, D.: Automatic classification of citation function. In: *Proceedings of the 2006 conference on empirical methods in natural language processing*. pp. 103–110. Association for Computational Linguistics (2006)
29. Witten, I.H., Frank, E., Hall, M.A.: *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edn. (2011)
30. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*. pp. 133–138. Association for Computational Linguistics (1994)