LaSTUS/TALN+INCO @ CL-SciSumm 2018 - Using Regression and Convolutions for Cross-document Semantic Linking and Summarization of Scholarly Literature

Ahmed Abura'ed[1], Àlex Bravo[1],
Luis Chiruzzo[2], and Horacio Saggion[1]

[1] Universitat Pompeu Fabra, DTIC, LaSTUS-TALN, C/Tanger 122, Barcelona (08018), Spain
first.last@upf.edu
[2] Universidad de la República, Facultad de Ingeniería, INCO, Montevideo, Uruguay
luischir@fing.edu.uy

Abstract. In this paper we present several systems developed to participate in the 3rd Computational Linguistics Scientific Document Summarization Shared challenge which addresses the problem of summarizing a scientific paper taking advantage of its citation network (i.e., the papers that cite the given paper). Given a cluster of scientific documents where one is a reference paper (RP) and the remaining documents are papers citing the reference, two tasks are proposed: (i) to identify which sentences in the reference paper are being cited and why they are cited, and (ii) to produce a citation-based summary of the reference paper using the information in the cluster. Our systems are based on both supervised (Convolutional Neural Networks) and unsupervised techiques taking advantage of word embeddings representations and features computed from the linguistic and semantic analysis of the documents.

Keywords: Citation-based Summarization · Scientific Document Analysis · Convolutional Neural Networks · Text-similarity Measures

1 Introduction

Automatic text summarization [28] is a technology to produce an abridged version of a document or set of documents which should contain just the essential or most relevant information of the input. Summarization of scientific documents has been studied for many years now (e.g. [19] was the first to address summarization of technical articles) and has been addressed with a variety of techniques including trainable [32] or dictionary-based [31] approaches. In recent years there has been a growing interest in the problem of citation-based summarization [25, 2, 1] where given a cluster of scientific documents in which one document in the cluster is a reference paper and the rest of the documents are papers citing the

reference paper, the objective is to summarize the reference paper taking in consideration the information citing it. The interest in the area has motivated the development of a series of evaluation exercises in scientific summarization in the Computational Linguistics (CL) domain known as the Computational Linguistics Scientific Document Summarization Shared Task which started in 2014 as a pilot [12] and which is now a well depeloped challenge in its third year [10, 11].

In this challenge, given a cluster of n documents where one is a reference paper (RP) and the $n-1$ remaining documents are papers (i.e., citing papers (CPs)) citing the reference paper, participants of the challenge have to develop automatic procedures to simulate 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 this system description paper, we overview the techniques we have applied to solve the proposed tasks.

2 Corpus Processing

The organizers of the CL-SciSumm 2018 challenge provide training data structured in clusters of reference and citing papers together with manual annotations indicating, for each citance, the text span(s) in the reference paper that best represent the citance, as well as their corresponding facets. The training corpus contains 40 clusters with an average of 17 papers per cluster. For each cluster there are three manually created summaries of the reference paper: the author abstract, a community-based abstract created using citation sentences, and a human abstract created based on information from reference paper and citation sentences. The test set has 20 clusters with an average of 11 papers per cluster.

2.1 GATE Transformation

In order to automatically process the clusters, we created, from the documents in the training and test sets, GATE [21] files that include the information provided in the manual annotations. The files corresponding to reference papers were enriched with annotations covering the text spans being cited (with the information corresponding to citances) and, conversely, in each citing paper annotations were added for the provided citances (with the information corresponding to the cited text spans). The annotations in the citing and reference papers are linked by means of a unique identifier (formed by the concatenation of citance number, reference paper id, citing paper id, and annotator).

Based on these annotations we could easily build pairs of matching sentences *(Citing Paper Sentence, Reference Paper Sentence)* and associate, to each pair, the facet that the annotator considered the citation referred to.

2.2 Text Processing

The GATE system was used to tokenize, sentence split, part of speech tag, manage gazetteers and lemmatize each document. Teufel's [33] action and concept Lexicons were used to create gazetteers lists to identify in text scientific concepts (e.g. *research*: analyze, check and gather; *problem*: violate, spoil and mistake, and *solution*: fix, cure and accomplish). The Dr Inventor's library [27] for analyzing scientific documents was additionally applied to each document to generate rich semantic information such as citation marker, BabelNet concepts [23], causality markers, co-reference chains, and rhetorical sentence classification. The library classifies each sentence of a paper based on a rhetorical category of scientific discourse among: Approach, Background, Challenge, Outcome and FutureWork. In other words, it predicts the probability of the sentence of belonging to one of the five discourses provided. See [8] for more details about the corpus used for training the classifier. Finally, the SUMMA library [29] was used to produce term vectors, normalized term vectors, BabelNet synset ID vectors, normalized babelnet synset ID vectors, terms n-grams (up to three) and part of speech n-grams (up to three) for each document.

3 Identifying Cited Sentences and their Facets

We utilized a deep-learning approach (Convolutional Neural Network (CNN)) formulating the problem of finding a set of sentences in a reference paper that best reflects a citation in a citing paper as a regression problem which uses a CNN with two inputs and one output. The first input models the reference paper sentences as a Word2Vec representation and the second input calculates a set of features based on the pair of sentences (reference paper sentence and a citation sentence). On the other hand, the regression output represents a score for each sentence in the reference paper based on the set of citations citing the reference paper. The output score of each reference sentence is based on the distance between the sentence and a sentence that are being cited. A value of 1 is set to cited sentences and the further the sentence is from the nearest cited sentence the less score it has.

We also used the same neural network to predict the facet a cited sentence belongs to. However, for facets we formulated the problem as a classification problem in which the output in that case is one of the five predefined facets classes provided by the organizers.

We modeled each reference sentence as a Word2Vec representation from three different pre-trained Word2Vec models embedded in a 300 dimensional space: (1)

ACL [3][18] from the ACL Anthology Reference Corpus [6], (2) Google News [4], and (3) Babelnet [7].

From each reference paper we extracted all the sentences having a number of tokens in a range of 5 to 40 and we used the 300 dimensions of each of the first 15 tokens from each sentence. In order to reduce the number of pairs of sentences to consider, we also excluded sentences which according to the analysis carried out with the Dr. Inventor library belongs to the *Background* or *Future Work* discourse facets since it is assumed those sentences will mainly refer to work carried out by other authors or still inexistent.

We ran three CNNs over each sentence embeddings in which the width is the 300 dimensions, the height is 2, 3 and 4 respectively to represent: bi, tri and quadri-grams and finally, 3 channels to present the three pre-trained models.

3.1 Set of Features

In addition to CNNs we calculated a set of features based on the pair of the cited sentence in the reference paper and the sentence citing it in the citing paper. Those features were modeled and motivated to identify the cited sentences and their facets. **Sentence Position Features**: The sentence position in a paper can identify which parts of a reference paper are mostly cited, also the facet the sentence belongs to. For instance, sentences at the end of the document would probably belong to the *Result* facet. We use three features based on the location of the sentence in the reference document:

- Sentence position: the position of the sentence in the reference paper;
- Section sentence position: the position of the sentence in the section;

WordNet Semantic Similarity Measures Features: Similarities derived from WordNet's graph could indicate matched sentences, it could also identify their facets, the more similar a sentence is to another sentence with a known facet the more likely it will have the same facet. 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. 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. The computed measures are:

- Path similarity [9]: The shorter the path between two words/senses in WordNet, the more similar they are.
- JCN similarity [13]: the conditional probability of encountering an instance of a child-synset given an instance of a parent synset.

[3] https://github.com/liuhaixiachina/Sentiment-Analysis-of-Citations-Using-Word2vec/tree/master/trainedmodels

[4] https://code.google.com/archive/p/word2vec/

- LCH similarity [15]: the length of the shortest path between two synsets for their measure of similarity.
- LESK similarity [5]: Similarity of two concepts is defined as a function of the overlap between the corresponding definitions (i.e., their WordNet glosses).
- LIN similarity [17]: 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 [26]: The probability of encountering an instance of concept c in a large corpus.
- WUP similarity [34]: The depths of the two synsets in the WordNet taxonomies, along with the depth of the lowest common subsumer.

Text Similarity Features: The more similar a text is to another text the more likely it is citing it and they will be part of the same facet. We used two different tf*idf vector representations of the sentences produced by the SUMMA library–one based on word lemmas and one on BabelNet synsets–and computed their cosine similarity. We also calculated the Jaccard and Modified Jaccard coefficients for the lemmas, generating a total of four text similarity features.
Dr Inventor Sentence Related Features: Other features obtained by means of the DRI Framework that we believed could be of use in predicting a sentence belonging to a particular facet include:

- Citation marker: three features to represent the number of citation markers in the reference sentence, citing sentence and the pair of sentences together;
- Cause and effect: two features to represent if the reference or citing sentence participates in one or more causal relations;
- Co-reference chains: three features to represent the number of nominals and pro-nominals chained in the reference sentence, citing sentence and the pair of sentences together
- Rhetorical category with highest probability: We mentioned in Section 2.2 that the DRI Framework predicts the probability of a sentence being in one of five possible rhetorical categories. Even if they are different from our targeted discourse facets, we believe that these probabilities could be informative for our tasks therefore we included a feature indicating the rhetorical category with the highest probability.

Scientific Gazetteer Features: We generated two features based on Teufel's action and concept lexicon. The lexicon contains 58 lists. The features are computed based on the reference sentence.

3.2 Unsupervised Approaches

We performed experiments using two unsupervised methods. In both experiments, we compare all the sentences in the reference paper to the citation text using some distance metric, then we consider the closest sentences according to this metric as candidates. The two metrics defined are the following:

- **Modified Jaccard**: We used a metric similar to the Jaccard similarity coefficient for comparing the two sentences [4]. This metric considers the union and intersection of words (like the Jaccard coefficient) but uses the inverted frequency information to give more weight to words in the intersection that are less common.
- **BabelNet Embeddings**: We obtained the BabelNet synsets for both sentences and transformed them into synset embeddings [20]. We then take the cosine similarity between the centroids of the synset embeddings for both the reference and citation sentences.

The only parameter to adjust using these methods is the number of sentences to consider as candidates. In order to optimize this parameter, we tested against CL-SciSumm 2017 test data, which is also a subset of CL-SciSumm 2018 training data. The results of these experiments are shown in table 1. The best result is achieved using the BabelNet Embeddings metric, considering only the closest sentence as candidate. The best result for Modified Jaccard is also close in F-measure.

Table 1. Performance for Task 1A unsupervised approaches over the CL-SciSumm 2017 test set.

Method	#Sents	Precision	Recall	F-Measure
MJ	1	0.120	0.103	0.111
MJ	2	0.072	0.115	0.088
BN	1	0.123	0.105	**0.113**
BN	2	0.085	0.142	0.106

3.3 Voting scheme

We designed a voting scheme that intended to leverage the strengths of the different supervised and unsupervised approaches. We considered these four system runs for the voting scheme:

- Top 10 sentences from the Convolutional Neural Network using learning rate 0.0001.
- Top 10 sentences according to Modified Jaccard unsupervised approach.
- Top 10 sentences according to BabelNet Embeddings unsupervised approach.
- Top 40 sentences for each target paper according to the relevance scores described in section 4.2.

The voting scheme returns a candidate sentence if at least N of the four systems agree on that sentence. The results are ordered according to relevance score and if there are more than five candidates, only the top five are selected. If there are no candidates in the intersection, the top sentence according to the BabelNet embeddings approach is used as a fallback mechanism. We submitted two runs using $N = 2$ and $N = 3$.

4 Summarization of Scientific Articles

In this section, we describe our extractive text summarization approach based on convolutional neural networks which extends on our previous work on trainable summarization [30, 3]. The network generates a summary by selecting the most relevant sentences from the RP using linguistic and semantic features from RP and CPs. The aim of our CNN is to learn the relation between a sentence and a scoring value indicating its relevance.

4.1 Context Features

In order to extract the linguistic information from both sources (RP and CPs), we developed a complex feature extraction method to characterize each sentence in the RP and its relation with the corresponding CPs.

Before the extraction of context features, we compute word vectors based on Word Embeddings and term frequencies, as described above (see Section 3). Specifically we used Google News and ACL pre-trained word embeddings and term frequency vectors by SUMMA. Then, for each sentence in the RPs and CPs, we compute three sentence vectors based on the centroid (Google, ACL and SUMMA). In addition, for each RP, we also computed the three centroids based on the abstract and the whole article.

From these vector representations, we extracted a set of numeric features some of which are based on comparing a sentence to its (document or cluster) context:

- Sentence Abstract Similarity Scores: the similarity of a sentence vector to the author abstract vectors (three features).
- Sentence Centroid Similarity Scores: the similarity of a sentence vector to the article centroid (three features).
- First Sentence Similarity Scores: the similarity of a sentence vector to the vector of the first sentence, that is, the title of the RP (three features).
- Position Score: a score representing the position of the sentence in the article. Sentences at the beginning of the article have high scores and sentence at the end of the article have low scores.
- Position in Section Score: a score representing the position of the sentence in the section of the article. Sentences in first section get higher scores, sentences in last section get low scores.
- Position in a Specific Section Score: a score representing the position of the sentence in a particular section. Sentences at the beginning of the section get higher scores and sentences at the end of the section get lower scores.
- TextRank Normalized Scores: a sentence vector is computed to obtain a normalized score using the TextRank algorithm [22] (three features).
- Term Frequency Score: we sum up the tf*idf values of all words in the sentence. Then, the obtained value is normalized using the set of scores from the whole article.

- Citation Marker Score: the ratio of the number of citation markers in the sentence to the total number of citation markers in the article.
- Rhetorical Class Probability Scores: as described before (see Section 3.1).
- Citing Paper Maximum Similarity Scores: each RP sentence vector is compared to each citation vector in each CP to get the maximum possible cosine similarity (three features).
- Citing Paper Minimum Similarity Scores: each RP sentence vector is compared to each citation vector in each CP to get the minimum possible cosine similarity (three features).
- Citing Paper Average Similarity Scores: each RP sentence vector is compared to each citation vector and the average cosine value obtained (three features).

4.2 Scoring Values

As commented above, our CNN learns the relation between features and a score, that is, a regression task by devising various scoring functions to represent the likelihood of a sentence belonging to a summary (for abstract, community and human). The nomenclature followed to symbolize a scoring function is SC_{Sum}, where SC is the specific scoring function (which is indicated bellow) and Sum is any summary type: abstract (Abs), community (Com) or human (Hum). The scoring functions are defined bellow:

- Cosine Distance: we calculated the maximum cosine similarity between each sentence vector in the RP with each vector in the gold standard summaries. This method produced three scoring functions (SUMMA (SU_{Sum}), ACL (ACL_{Sum}), and Google (Go_{Sum})) for each summary type.
- ROUGE-2 Similarity: we also calculated similarities based on the overlap of bigrams between sentences in the RP and gold standard summaries. In this regard, each sentence in the RP is compared with each gold standard summary using ROUGE-2 [16]. The precision value from this comparison is taken for the scoring function and is symbolized as $R2_{Sum}$.
- Scoring Functions Average: Moreover, we computed the average between all scoring functions (SUMMA, ACL, Google and ROUGE-2) for each summary type. In addition, we also calculated a simplified average with vectors do not based on word-frequencies (ACL, Google and ROUGE-2). These scoring functions are indicated as Av_{Sum} and SAv_{Sum}, respectively.

Finally, these computation produced eighteen different functions to learn: SUMMA (SU), ACL (ACL) and Google (Go) vectors, ROUGE-2 ($R2$), Average (Av) and Simplified Average (SAv) times abstract (Abs), community (Com), human (Hum) summaries.

4.3 Convolution Model

Basically, a CNN consists of multiple convolutional and pooling layers, with fully-connected layers at the end. The network is fed with two different inputs. The

inputs are composed of instances related to sentences. The first one is based on the context features (described in the section 4.1). Specifically, *context features* are introduced in the CNN within a sequential window including the context features of the 3 previous and 3 following sentences. And the second input is related to the word embedding information for each sentence. In particular, we used both word embeddings (Google and ACL) as a dual channel, which stopwords were removed, the size was fixed in 15 words and they were kept static during the training.

Regarding the neural network hyperparameters, the CNN was defined with the Adadelta updater [35] and the gradients were computed using back-propagation as Kim [14] and Nguyen [24]. Also we used the sigmoid activation function, a dropout rate of 0.5, l2 constraint of 3. For the convolutions, we applied 3 filter window sizes (3, 4 and 5) to context features and 4 filter window sizes (2, 3, 4 and 5) to word embeddings. For each window were applied 150 filters for convolution. Finally, for learning the regression task we applied a Mean Squared Error (MSE) as loss function.

4.4 Evaluation

The evaluation consists of generating a 250-word summary according to the task, which are compared against each of the summary types of the gold standard: the reference papers abstract, a community summary, and a human summary. We trained and evaluated our model using the CL-SciSumm-17 dataset.

Table 2. ROUGE-2 and ROUGE-SU4 best results for each summary evaluation. In addition, the scoring function employed is specified under each value. The results are based on the F-score value.

Method	ROUGE-2			ROUGE-SU4		
	Abstract	Community	Human	Abstract	Community	Human
Winning Score 2017	0.351	0.217	0.275	0.191	0.174	0.178
Dual Channel	0.555	0.274	0.288	0.290	0.193	0.240
Scoring Function	SAv_{Abs}	Av_{Hum}	$R2_{Abs}$	SAv_{Abs}	Av_{Hum}	Go_{Abs}

Table 2 shows the winner scores achieved by the participants in the CL-SciSumm-17 Shared Task 2 (in the first row) and also shows the most representative results achieved in our experiments (in the second). The values shown are also based on the F-score obtained in both ROUGE-2 and ROUGE-SU4 evaluations.

5 Challenge Submissions

We have submitted the following systems for Task 1 (a and b):

- MJ1: unsupervised approach using Modified Jaccard similarity

- BN1: unsupervised approach using BabelNet synset embeddings cosine similarity
- 0.1CNN4: deep learning approach using CNN over the word embedding + a set of features . Learning rate: 0.1 Epoch: 50
- 0.0001CNN4: deep learning approach using CNN over the word embedding + a set of features . Learning rate: 0.0001 Epoch: 50
- Voting2: keep candidates if at least two of the systems agree (MJ, BN, CNN or top 40 sentences using task2 system)
- Voting3: keep candidates if at least three of the systems agree (MJ, BN, CNN or top 40 sentences using task2 system)

For the task 2, we have submitted eighteen summaries related to each scoring function and summaries. In other words, each resulting summary is defined by SC_{Sum} (see Section 4.2), where SC is the scoring function (SU, ACL, Go, $R2$, Av and SAv) and Sum is the summary type (Abs, Com and Hum). For example, submission ACL_abs learns a scoring function which attempts to approximate similarity of a sentence to the abstract of the document using ACL vectors and cosine to compute similarities.

6 Conclusion

We have described the systems developed to participare in Tasks 1a, 1b and 2 in the CL-SciSumm 2018 summarization challenge. For Task 1a – which aimed at identifying cited sentences –, we implemented supervised and unsupervised methods. Our supervised systems are based on Convolutional Neural Networks (CNN), while the unsupervised techniques take advantage of word embedding representations and features computed from the linguistic and semantic analysis of the documents. However, as committing to only one system could result in an underperforming approach, we applied many different system configurations combining them through a voting mechanism. For Task 1b we used the same CNN system of Task 1a where the output was a set of facets.

Regarding Task 2 – summarization proper –, we have developed a neural network based on convolutions to learn a specific scoring function. The CNN model was fed by a combination of word embeddings with sentence relevance and citation features extracted from each document cluster (RP and CPs). The approach was developed and evaluated following the CL-SciSumm Shared Task 2 dataset, our approach outperformed results reported in last year CL-SciSumm-17 Shared Task 2.

Acknowledgments

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) and by the TUNER project (TIN2015-65308-C5-5-R, MINECO/FEDER, UE).

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. AbuRa'ed, A., Chiruzzo, L., Saggion, H., Accuosto, P., Bravo, À.: Lastus/taln @ clscisumm-17: Cross-document sentence matching and scientific text summarization systems. 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) and co-located with the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017), Tokyo, Japan, August 11, 2017. pp. 55–66 (2017)
4. AbuRaed, A., Chiruzzo, L., Saggion, H.: What sentence are you referring to and why? identifying cited sentences in scientific literature. In: RANLP 2017. International Conference Recent Advances in Natural Language Processing; 2017 Sep 2-8; Varna, Bulgaria.[Stroudsburg (PA)]: ACL; 2017. p. 9-17. ACL (Association for Computational Linguistics) (2017)
5. Banerjee, S., Pedersen, T.: An adapted lesk algorithm for word sense disambiguation using wordnet. In: International Conference on Intelligent Text Processing and Computational Linguistics. pp. 136–145. Springer (2002)
6. Bird, S.: The ACL anthology reference corpus: A reference dataset for bibliographic research in computational linguistics (2008)
7. Camacho-Collados, J., Pilehvar, M.T., Navigli, R.: Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. Artificial Intelligence **240**, 36–64 (2016)
8. Fisas, B., Ronzano, F., Saggion, H.: A multi-layered annotated corpus of scientific papers. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016. (2016)
9. 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)
10. 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)
11. 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)
12. 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)
13. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. arXiv preprint cmp-lg/9709008 (1997)
14. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)

15. Leacock, C., Chodorow, M.: Combining local context and wordnet similarity for word sense identification. WordNet: An electronic lexical database **49**(2), 265–283 (1998)
16. Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Text summarization branches out: Proceedings of the ACL-04 workshop. vol. 8. Barcelona, Spain (2004)
17. Lin, D.: An information-theoretic definition of similarity. In: ICML. vol. 98, pp. 296–304. Citeseer (1998)
18. Liu, H.: Sentiment analysis of citations using word2vec. arXiv preprint arXiv:1704.00177 (2017)
19. Luhn, H.P.: The automatic creation of literature abstracts. IBM J. Res. Dev. **2**(2), 159–165 (Apr 1958)
20. Mancini, M., Camacho-Collados, J., Iacobacci, I., Navigli, R.: Embedding words and senses together via joint knowledge-enhanced training. arXiv preprint arXiv:1612.02703 (2016)
21. Maynard, D., Tablan, V., Cunningham, H., Ursu, C., Saggion, H., Bontcheva, K., Wilks, Y.: Architectural elements of language engineering robustness. Natural Language Engineering **8**(2-3), 257–274 (2002)
22. Mihalcea, R., Tarau, P.: Textrank: Bringing order into text. In: Proceedings of the 2004 conference on empirical methods in natural language processing (2004)
23. Navigli, R., Ponzetto, S.P.: BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artif. Intell. **193**, 217–250 (Dec 2012)
24. Nguyen, T.H., Grishman, R.: Relation extraction: Perspective from convolutional neural networks. In: Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing. pp. 39–48 (2015)
25. Qazvinian, V., Radev, D.R.: Scientific paper summarization using citation summary networks. In: Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1. pp. 689–696. COLING '08, Association for Computational Linguistics, Stroudsburg, PA, USA (2008)
26. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. arXiv preprint cmp-lg/9511007 (1995)
27. Ronzano, F., Saggion, H.: Dr. Inventor Framework: Extracting structured information from scientific publications. In: International Conference on Discovery Science. pp. 209–220. Springer (2015)
28. Saggion, H., Poibeau, T.: Automatic text summarization: Past, present andfuture. In: Poibeau, T., Saggion, H., Piskorski, J., Yangarber, R. (eds.) Multi-source, Multilingual Information Extraction and Summarization. Springer Verlag, Berlin (2013)
29. Saggion, H.: SUMMA: A robust and adaptable summarization tool. Traitement Automatique des Langues **49**(2), 103–125 (2008)
30. Saggion, H., AbuRa'ed, A., Ronzano, F.: Trainable citation-enhanced summarization of scientific articles. In: Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL) co-located with the Joint Conference on Digital Libraries 2016 (JCDL 2016), Newark, NJ, USA, June 23, 2016. pp. 175–186 (2016)
31. Saggion, H., Lapalme, G.: Generating indicative-informative summaries with sumum. Comput. Linguist. **28**(4), 497–526 (Dec 2002)
32. Teufel, S., Moens, M.: Summarizing scientific articles: Experiments with relevance and rhetorical status. Comput. Linguist. **28**(4), 409–445 (Dec 2002)

33. Teufel, S., et al.: Argumentative zoning: Information extraction from scientific text. Ph.D. thesis, Citeseer (2000)
34. 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)
35. Zeiler, M.D.: Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)