Scientometrics manuscript No.
(will be inserted by the editor)

Automatic Related Work Section Generation: Experiments in Scientific Document Abstracting

**Ahmed AbuRa'ed · Horacio Saggion ·
Alexander Shvets · Àlex Bravo**

Received: date / Accepted: date

Abstract Related work sections or literature reviews are an essential part of every scientific article being crucial for paper reviewing and assessment. However, writing a good related work section is an activity which requires considerable expertise to identify, condense/summarize, and combine relevant information from different sources. In this work we compare different automatic methods to produce "descriptive" related work sections given as input the set of papers which need to be described. The main contribution of our work is a neural sequence learning process which produces citation sentences to be included in a related work section of an article. We train the neural architecture using an available scientific data set of citation sentences and we test over a data set of related work sections; we also compare the performance to a set of baseline extractive summarizers, an abstractive summarizer and a state of the art CNNs approach. Our results indicate that our approach outperforms the simple as well as the informed baselines.

Article Highlights

- The design and evaluation of an abstractive related work section generation system;

Ahmed AbuRa'ed
Universitat Pompeu Fabra, Barcelona, Spain
E-mail: ahmed.aburaed@upf.edu

Horacio Saggion
Universitat Pompeu Fabra, Barcelona, Spain
E-mail: horacio.saggion@upf.edu

Alexander Shvets
Universitat Pompeu Fabra, Barcelona, Spain
E-mail: alexander.shvets@upf.edu

Àlex Bravo
Universitat Pompeu Fabra, Barcelona, Spain
E-mail: alex.bravo@upf.edu

- A new data set of pairs of articles and citation sentences to train sequence-to-sequence models;
- A comparison with state-of-the-art methods showing the potential of the approach.

Keywords Scientific Summarization · Document Abstracting · Sequence Learning · Information Extraction from Scientific Literature

Mathematics Subject Classification (2010) 68T50 · 97R40

CR Subject Classification I.2.7

1 Introduction

Every scientific paper should include a related work section providing, in a well organized and condensed form, the key information from a carefully selected list of publications which contextualize and ground the research being presented by an author (Rowley and Slack, 2004). Moreover, their relevance is critical for quality assessment since journals pay particular attention to related work sections where evaluation of manuscripts is of concern (Maggio et al., 2016). The reason why a scientific paper should include a related work section is motivated by the fact that scientific research is a collective activity. The work of researchers depends on knowledge accumulated by scientists and scholars over years of research. Therefore, an author often needs to describe related previous works for the readers to help them understand the context of his or her contributions in an area of research, also facilitating any form of comparison between the current and previous works. Good related work sections are difficult to produce since they require the author to select, contrast, and organize key information from several sources. It is generally agreed that related work sections or literature reviews can either be **descriptive** or **integrative** (S. G. Khoo et al., 2011; Jaidka et al., 2013). While a descriptive report will summarize individual papers providing information such as methods and results in citation sentences, integrative reports will focus on key ideas and topics, providing in the citation sentences critical views on the presented approaches. In a context where scientific information is growing at an unprecedented pace, related work sections or literature reviews offer already digested information ready to be used by researchers interested in getting a gist of the state of the art. Automatically generating this type of text, that is selecting and combining key information from a set of articles, could greatly help researchers in coping with the problem of scientific information overload.

In this paper we are concerned with the automatic production of descriptive related work sections from a set of selected papers. We do not attempt to generate integrative reviews since they will require knowledge difficult to encode in an automatic process. Moreover, recommending a pre-selection set of scientific papers to be included in the report is outside the scope of this paper. To further investigate possible ways of compiling a list of scientific papers to cite, see (McNee et al., 2002).

Past research has shown that descriptive related work sections usually bring and combine information from titles, abstracts, and introductions of the cited articles making use of cut-and-paste summarization strategies (Jaidka et al., 2013) which are typical of abstracting a document (insertion, deletion, substitution, etc.) (Endres-Niggemeyer et al., 1995; Saggion, 2011). These observations motivate our generative approach to the automatic production of related work sections.
Taking advantage of an available data set for scientific summarization composed of research articles, citation sentences, and human summaries we train a sequence-to-sequence model to simulate the generation of citation sentences. We concatenate citation sentences automatically generated from each cited paper to produce a novel related work section which we evaluate by comparing the generated texts to the gold related work section using content-based evaluation metrics. The comparison is carried out with our abstractive approach, several baselines, unsupervised summarizers, and an extractive state of the art neural networks approach.

To model our generative approach, we make use of pointer–generator neural networks (Vinyals et al., 2015a) which are sequence-to-sequence models that produce an output sequence consisting of elements from the input sequence. We use the pointer–generator networks with two Neural Networks (NN) architectures which have recently achieved good performance in complicated tasks; Transformer (Vaswani et al., 2017), that uses stacked self-attention and point-wise fully connected layers for both the encoder and decoder, and Bi-Directional RNNs, more specifically, in (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Cho et al., 2014) it is introduced as a variation of RNNs called sequence-to-sequence (seq2seq) learning which uses recurrent neural networks to map variable-length input sequences to variable-length output sequences. While relatively new, the sequence-to-sequence approach has achieved state-of-the-art results in not only its original application – machine translation – (Luong and Manning, 2015; Jean et al., 2014; Luong et al., 2015; Jean et al., 2015; Luong et al., 2014), but also image caption generation (Vinyals et al., 2015b), and text summarization (Nallapati et al., 2016).

Sequence-to-sequence learning aims to indirectly model the conditional probability $p(y|x)$ of mapping an input sequence, $x = x_1, ..., x_n$, into an output sequence, $y = y_1, ..., y_m$ accomplishing such goal through the encoder-decoder framework proposed by (Sutskever et al., 2014; Cho et al., 2014). We use sequence-to-sequence architecture to generate each citation sentence to be included in the related work section from an input sequence which is composed of a title and an abstract of a scientific paper that is being cited.

To directly tackle the problem of producing a related work section, we use a gold-standard data set of related work sections and their cited papers to test our approach. We feed our model with a set of sentences from the cited papers and accumulate the generated citation sentences to produce a related work section.

The contributions of our work are the following:

- The design and evaluation of an abstractive related work section generation system;
- A new data set of over 15K pairs of articles and citation sentences to train sequence-to-sequence models;
- A comparison with state-of-the-art methods showing the potential of the approach.

Software and data are being made available to guarantee reproducible research [1]. The rest of the paper is organized as follows: In the next Section we give an overview of related work in the broader area of scientific text summarization and in the more focused problem of related work section generation. Then in Section 3 we describe the data sets created and/or used in our experiments while in Section 4 we describe the method used for our sequence-to-sequence system. Section 5 describes the experiments carried out, Section 6 presents the obtained results while in Section 7 limitations of the study are highlighted. Finally, we close the paper in Section 8 with conclusions and avenues for further research.

2 Related Work

Related work for our research refers to the broad topic of scientific text summarization as well as to the more targeted generation of related work reports. Summarization of scientific and technical articles has been studied for a long time (Saggion and Poibeau, 2013). Early approaches to single document summarization of scientific input has been addressed with sentence classification (Teufel and Moens, 2002), domain specific pattern-based matching and extraction (Oakes and Paice, 1999), or generic information extraction and text generation techniques (Saggion and Lapalme, 2002). More recently, multi-document summarization of scientific texts took center stage. (Agarwal et al., 2011) tackled the multi-document summarization of scientific articles using an unsupervised method which discovers comparable attributes in co-cited articles using Frequent Term Based Clustering (Beil et al., 2002). Discovered clusters are used to rank and extract sentences for the summary. Qazvinian et al. (2013) proposed C-LexRank, a graph-based summarization method which relies on implicit as well as explicit citation sentences to summarize a given cited paper. They cluster the citation sentences extracting the most relevant from each cluster using different procedures. Jha et al. (2013) implemented a simliar system but to generate a survey of a given topic. Their approach identifies different aspects of the scientific paper extracting representative sentences for each aspect. Mohammad et al. (2009) performed experiments to show the helpfulness of citation text to automatically generate technical surveys while

[1] `https://github.com/AhmedAbuRaed/SPSeq2Seq`

(Ronzano and Saggion, 2016) using data from the BioSumm 2014 Challenge studied performance gains when using citation sentences to summarize a scientific article.

Recent approaches to abstractive summarization include the following. Bražinskas et al. (2019) has addressed opinions summarization in which they analyze multiple reviews from users over different products and businesses and then created text summaries that reflect subjective information expressed in these reviews. To overcome any rely on large quantities of document-summary pairs as used in supervised abstractive summarization which are expensive to acquire, they used an unsupervised approach which uses a hierarchical variational auto-encoder (VAE) model and utilizes two sets of latent variables. A continuous variable that captures latent semantics of a group of reviews and a second continuous variable to encode latent semantics of each individual review in the group. The final summaries are produced by the decoder that uses the information stored at the second continues variable. Chu and Liu (2018) also utilized an unsupervised abstractive summarization model that uses an autoencoder where the mean of the representations of the input reviews (i.e. mean over the hidden and cell states of all the input reviews) decodes to a reasonable summary-review while not relying on any review-specific features. They implemented variants of the proposed architecture and analyzed the different variants. Finally, Baziotis et al. (2019) also uses an unsupervised abstractive model to develop a sequence-to-sequence-to-sequence autoencoder (SEQ^3), where the first sequence is the input, the second sequence is the compressed sentence and the last sequence consists of reconstructed sentences. SEQ^3 consists of two chained encoder-decoder pairs, with words used as a sequence of discrete latent variables.

In contrast with generic summarization, related work section generation - summarization has not been so extensively explored. Hoang and Kan (2010); Vu (2010) presented an automatic related work summarization system which creates a topic-biased related work section for a target paper given multiple scientific articles. The extractive approach requires the user to provide a topic hierarchy tree as an input and a set of papers to summarize. The method, which improves over generic multi-document summarization approaches, computes the likelihood of each sentence in the input documents to belong to the topics as a method for selection.

Hu and Wan (2014) investigated the task of producing a related work section for a target paper given as input a set of reference papers along with a target academic paper but ignoring its related work section. Their system exploits Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 1999) to divide the set of sentences of the given papers into different topic-biased parts applying regression models to learn the ranking of the sentences. Finally, an optimization framework is used to produce the related work section. In order to reduce the amount of text to consider, they make use of the abstract, introduction, related work and conclusion sections from the reference papers, since according to them other sections corresponding to method, evaluation or any other sections always describe in too much details any specific work and they are not suitable for this task. Jaidka et al. (2013) established a literature review framework by deconstructing human-written literature review sections of information science research papers. They studied scientific papers to be able to compare them, to identify new problems, to place a work inside the current literature and to elaborate new research propositions. Their study offers the results of a multi-level discourse analysis to examine their discourse and content features. A framework for literature reviews created focusing on macro-level document structure, sentence-level templates, and information summarization strategies. Zhang et al. (2018) proposed a latent variable extractive model that views labels of sentences in a document as binary latent variables. The latent model maximizes the likelihood of human summaries given selected sentences where loss comes directly from gold summaries. They modeled instances of sequence labeling in which a document is viewed as a sequence of sentences and the model is expected to predict a *true* or *false* label for each sentence, where *true* indicates that the sentence should be included in the summary. Their system has three parts: a sentence encoder to convert each sentence into a vector, a document encoder to learn sentence representations given surrounding sentences as context, and a document decoder to predict sentence labels based on representations learned by the document encoder. Finally, they use CNN/Dailymail data set (Hermann et al., 2015) for their experiments and they compare their system with other extractive and abstractive systems. Lastly, an hybrid method for summarization of multiple related work sections of scientific articles has recently been proposed (Altmami and Menai, 2018). In this work a semantic graph-based approach is used to handle the redundancy of citation sentences by reducing the sentence graph while preserving its properties. Using cross-document structure theory (CST) to analyze multi-documents i.e. related work section, they discover semantic relations to further reduce redundancy in the set of citation sentences.

Most reviewed approaches to related work section generation are based on an extractive paradigm. Extractive approaches, while offering the advantages of producing readable sentences, are clearly limited to address the challenges of producing citation sentences which are generally non-literal versions of information found in the input document. Moreover, citations sentences sometimes combine fragments from different sentences which can not be dealt with extractive approaches. These limitations could be addressed by applying non-extractive techniques as the ones we present in the rest of this article, which although still preliminary can pave the way for further research in this area.

3 Data

We make use of two different types of data: a data set of scientific papers and their citation sentences that we use to train our citation sentence generation model, and a gold-standard data set of related work sections and their cited papers to test the whole process. The testing data set has been used in previous work (Hoang and Kan, 2010; Vu, 2010), and has been considerably expanded by AbuRa'ed et al. (2020) who processed the data set and manually mapped the sentences of the cited papers with the sentences in related work sections citing them[2]. Additionally, we study the effect of a filter over the data sets in order to select sentences which explicitly indicate the author's work .

3.1 Training Datasets

We make use of the data available in the ScisummNet Corpus (Yasunaga et al., 2019b,a). This corpus is being released by Yale LILY lab and expanded from the CL-Scisumm project (Mayr et al., 2019; Jaidka et al., 2014). This dataset provides over 1,000 papers of the Association for Computational Linguistics (ACL) anthology network (Bird, 2008) with their citation networks (e.g. citation sentences, citation counts) and their author abstracts.

Additionally, we collect data similar to ScisummNet but from Open Academic Graph (OAG) and Microsoft Academic Graph (MAG) (Sinha et al., 2015; Tang et al., 2008). MAG is a diverse graph containing scientific publication records, citation relationships between those publications, as well as authors, institutions, journals, conferences, and fields of study. OAG is a large knowledge graph unifying two billion records from two academic graphs: Microsoft Academic Graph (MAG) and AMiner (Tang, 2016). We used the available OAG dumps to gain access to the list of all paper IDs at MAG. Afterward, we used Microsoft Cognitive Services Academic Knowledge API to access MAG nodes. The obtained papers were kept if and only if: (i) MAG contained an abstract for the paper and (ii) MAG contained at least one of the papers being cited. The references of the stored papers were extracted to iteratively obtain more data, storing papers only if conditions (i) and (ii) were satisfied. All the collected data, which will be released with this paper[3], has been indexed for efficient processing. The collected data amounts to: 940 pairs from ScisummNet Corpus and 15,574 pairs from our new dataset.

In summary, our data from these two sources consists of pairs of input and output sequences as follows:

$$< T_i \oplus A_i, C_i > \tag{1}$$

Where the $i-$th input sequence is a concatenation ($\oplus$) of a scientific paper's title T_i and abstract A_i, as for the output sequence we use the citation sentence C_i used by the citing scientific paper.

For further analysis we also applied a filter on the same data which selects sentences from the abstract that are directly related to the scientific paper

[2] http://taln.upf.edu/sciencecorpus

[3] The dataset can be accessed though this link: `https://github.com/AhmedAbuRaed/SPSeq2Seq`

author or presentation. The filter is based on Teufel's (Teufel et al., 2000) first pronoun (e.g. we, our and my) and presentation nouns (e.g. this paper, study and article) gazetteers. The filter is only applied to the abstract sentences (the title is never removed). The resulting sentences from the filter process are the title and abstracts' sentences that contain any of the first pronoun and presentation nouns. This process will exclude any sentences that do not explicitly mention the authors nor the presented work directly.

An example of the data used for training the citation sentence generator is shown in Figure 1. In the example[4], the citation sentence contains some literal (e.g. "negative evidence from edited textual corpora") and non-literal (e.g. "high precision" instead of "80% precision" or "checkers" instead of "detecting grammatical errors") elements extracted from title and abstract of the cited work. From the set of citation sentences available for each paper we use the one that is most similar (closest) to title and abstract in terms of the BLEU score measure (Papineni et al., 2002) used to compare a target and source translations.

title	*An* **Unsupervised** *Method For* **Detecting Grammatical Errors**
abstract	We present an **unsupervised** method for **detecting grammatical errors** by inferring **negative evidence from edited textual corpora**. The system was developed and tested using essay-length responses ... The error-recognition system, ALEK, performs with about **80% precision** and **20% recall**.
cit. sent.	Among **unsupervised checkers**, Chodorow and Leacock (2000) exploits **negative evidence from edited textual corpora** achieving **high precision** but **low recall.**

Fig. 1 Example of a scientific article (title ⊕ (non-filtered) abstract) and a citation sentence. Similar phrases have been highlighted.

A similar example showing the filtered version of the previous example can be seen in Figure 2. We can notice that after applying the filter process most of the shared phrases are still present.

title	*An* **Unsupervised** *Method For* **Detecting Grammatical Errors**
abstract	We present an **unsupervised** method for **detecting grammatical errors** by inferring **negative evidence from edited textual corpora**.
cit. sent.	Among **unsupervised checkers**, Chodorow and Leacock (2000) exploits **negative evidence from edited textual corpora** achieving high precision but low recall.

Fig. 2 Example of a Filtered scientific article (title ⊕ filtered abstract) and a citation sentence. Similar phrases have been highlighted.

[4] Cited paper: Martin Chodorow and Claudia Leacock. 2000. An unsupervised method for detecting grammatical errors. In Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference (NAACL 2000). Association for Computational Linguistics, Stroudsburg, PA, USA, 140-147. Citing paper: Chung-Chi Huang, Mei-Hua Chen, Shih-Ting Huang, Jason S. Chang. EdIt: A Broad-Coverage Grammar Checker Using Pattern Grammar. Proceedings of the ACL-HLT 2011 System Demonstrations.

3.2 Testing Data set

In order to test our approach, we make use of a data set previously used for related work generation (Hoang and Kan, 2010; Vu, 2010), the RWSData corpus, which contains a collection of 20 article sets (i.e. **Clusters**). Each set contains a scientific paper with a related work section, the text of the related work section and the set of reference papers mentioned in the related work section, See Figure 3 which represents a segment of a related work section for a scientific paper in the corpus [5] that is citing three different scientific papers [6] [7] [8].

Since the papers provided in RWSData are all in PDF and in order to extract the necessary information from them, AbuRa'ed et al. (2020) used three state-of-the-art PDF to XML converters: PDF Digest (Ferrés et al., 2018), PDFX (Constantin et al., 2013) and Grobid (GRO, 2008 — 2019). Whenever one of the converters failed to convert the PDF document we moved to next one as a fail safe, using this procedure allowed us to convert all the documents in the dataset and extract the necessary information for testing our system. The three converters provide basic information about each scientific paper including: title, abstract and content. Then, using the GATE system (Maynard et al., 2002) we automatically annotated (and manually checked) each citation sentence in the related work section of the target scientific paper linking it with its cited paper. Finally, the same filtering process applied on the training data set was applied for the testing data set.

4 Methodology

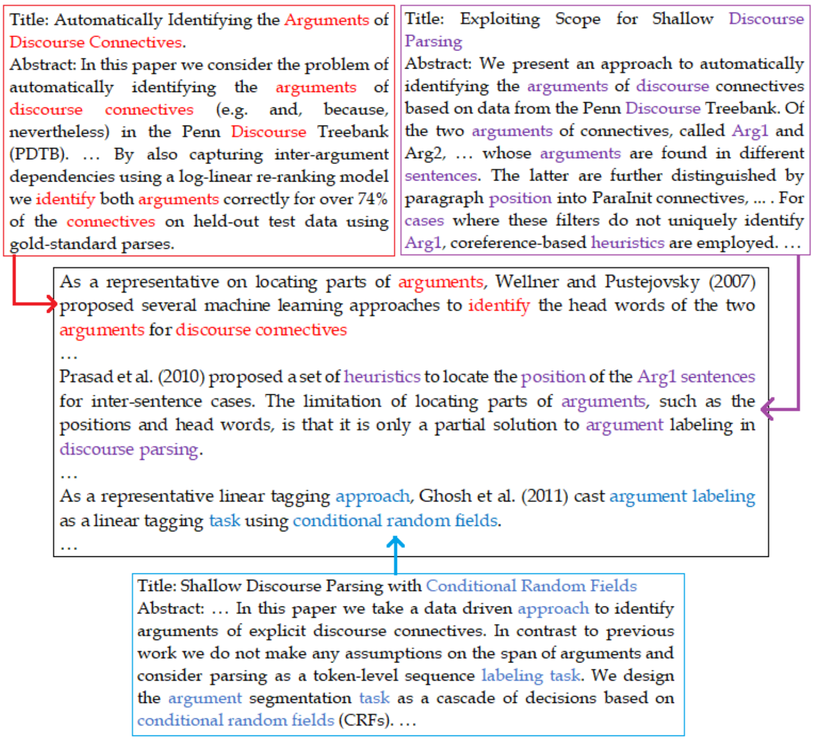
Our approach is based on pointer–generator neural networks with copy-attention technique and coverage mechanism (See et al., 2017; Wu et al., 2016). Copy-based generation can copy words from the source text via pointing, which aids accurate reproduction of information while retaining the ability to produce novel words through the generator. As for coverage (Wu et al., 2016), it is a mechanism to keep track of what has been summarized discouraging repetition by forcing penalties on repeated text therefore controlling redundancy of the generated output.

[5] Kong, Fang, Hwee Tou Ng, and Guodong Zhou. "A constituent-based approach to argument labeling with joint inference in discourse parsing." In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 68-77. 2014.

[6] Ben Wellner and James Pustejovsky. 2007. Automatically identifying the arguments of discourse connectives. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 92–101.

[7] Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. 2005. Attribution and the (non-)alignment of syntactic and discourse arguments of connectives. In Proceedings of the Workshop on Frontiers in Corpus Annotation II: Pie in the Sky, pages 29–36

[8] Sucheta Ghosh, Richard Johansson, Giuseppe Riccardi, and Sara Tonelli. 2011. Shallow discourse parsing with conditional random fields. In Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 1071–1079.



Title: Automatically Identifying the Arguments of Discourse Connectives.
Abstract: In this paper we consider the problem of automatically identifying the arguments of discourse connectives (e.g. and, because, nevertheless) in the Penn Discourse Treebank (PDTB). … By also capturing inter-argument dependencies using a log-linear re-ranking model we identify both arguments correctly for over 74% of the connectives on held-out test data using gold-standard parses.

Title: Exploiting Scope for Shallow Discourse Parsing
Abstract: We present an approach to automatically identifying the arguments of discourse connectives based on data from the Penn Discourse Treebank. Of the two arguments of connectives, called Arg1 and Arg2, … whose arguments are found in different sentences. The latter are further distinguished by paragraph position into ParaInit connectives, … . For cases where these filters do not uniquely identify Arg1, coreference-based heuristics are employed. …

As a representative on locating parts of arguments, Wellner and Pustejovsky (2007) proposed several machine learning approaches to identify the head words of the two arguments for discourse connectives
…
Prasad et al. (2010) proposed a set of heuristics to locate the position of the Arg1 sentences for inter-sentence cases. The limitation of locating parts of arguments, such as the positions and head words, is that it is only a partial solution to argument labeling in discourse parsing.
…
As a representative linear tagging approach, Ghosh et al. (2011) cast argument labeling as a linear tagging task using conditional random fields.
…

Title: Shallow Discourse Parsing with Conditional Random Fields
Abstract: … In this paper we take a data driven approach to identify arguments of explicit discourse connectives. In contrast to previous work we do not make any assumptions on the span of arguments and consider parsing as a token-level sequence labeling task. We design the argument segmentation task as a cascade of decisions based on conditional random fields (CRFs). …

Fig. 3 Example from the RWSData corpus of a scientific paper citing three other scientific papers.

We utilize pointer–generator neural networks with two different architectures; Bidirectional Recurrent Neural Networks (BRNN) (Schuster and Paliwal, 1997) which maps a source sequence to a target sequence, and Transformers (Vaswani et al., 2017); where the closest model to the one we use is so-called CopyTransformer proposed in (Gehrmann et al., 2018). See Figure 4 which shows the pointer–generator neural network used with the BRNN architecture. For each decoder time-step a generation probability $P_{gen} \in [0,1]$ is calculated, which weights the probability of generating words from the vocabulary, versus copying words from the source text. The vocabulary distribution and the attention distribution are weighted and summed to obtain the final distribution, from which we make our prediction. The figure presents an example of a scientific paper at the input text [9] being cited by another scientific paper [10] and the network is trying to generate the next token for the citation

[9] Cited paper: Brill, Eric. "A simple rule-based part of speech tagger." In Proceedings of the third conference on Applied natural language processing, pp. 152-155. Association for Computational Linguistics, 1992

[10] Citing Paper: Modi, Deepa, and Neeta Nain. "Part-of-Speech Tagging of Hindi Corpus Using Rule-Based Method." In Proceedings of the International Conference on Recent

context (summary) in which the next token to be generated by the decoder is “speech” which has the highest attention in the attention distribution.

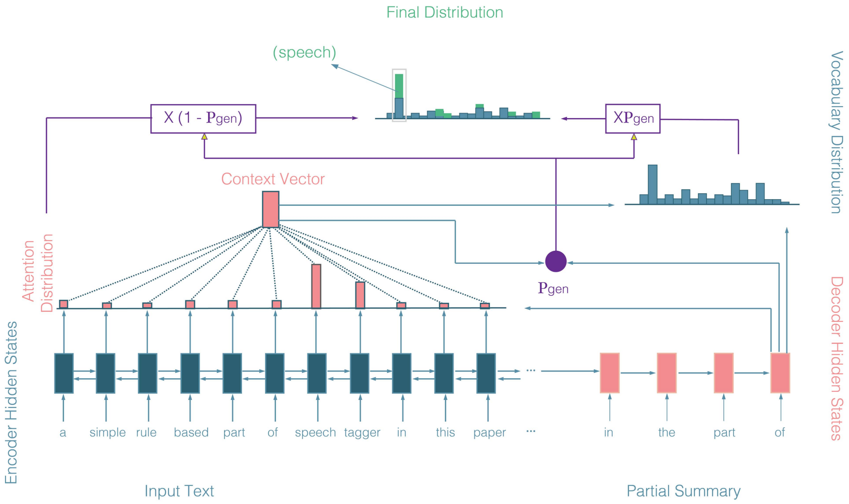


Fig. 4 The pointer–generator architecture.

Sequence-to-sequence models are particularly good at translation, where the sequence of words from one language is transformed into a sequence of different words in another language. However, summarization can, in certain cases, be casted as sequence-to-sequence modeling to summarize a long source into a shorter one in the same language to form the final output summary. We use BRNN (Schuster and Paliwal, 1997) as can be noticed at Figure 4 which is a natural generalization of feed-forward neural networks where the source sequence tokens are fed one-by-one into a single-layer of a bidirectional LSTM (encoder), producing a sequence of encoder hidden states h_i. On each step t, a single-layer of a unidirectional LSTM receives the word embedding of the previous word (while training, this is the previous word of the reference summary; at test time it is the previous word emitted by the decoder), and has decoder state st. We also applied *the transformer* (Vaswani et al., 2017) - encoder–decoder–based architecture - for ”translating” one sequence into another one as a basis. This architecture uses stacked self-attention and point-wise fully connected layers for both the encoder and decoder. See the model architecture at Figure 5 which we use with the pointer–generator neural network separately

Cognizance in Wireless Communication and Image Processing, pp. 241-247. Springer, New Delhi, 2016.

replacing the BRNN architecture. The encoder is composed of a stack of N identical layers. Each layer has two sub-layers. The first is a multi-head, self-attention mechanism, and the second is a simple, position wise fully connected feed-forward network. The decoder is also composed of a stack of N identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack. The transformer uses a self-attention layer by adding a mechanism called ”multi-headed” attention expanding the model's ability to focus on different positions of the input, giving the attention layer multiple ”representation subspaces” for the weight matrices, and allowing selection of important parts of the sequence at each step to adjust the distribution over the vocabulary which is essential while summarizing.
We rely on the Neural Machine Translation (NMT) tool OpenNMT-py (Klein et al., 2017) to implement our abstractive models. OpenNMT is an open source initiative for NMT and neural sequence modeling. It is a general-purpose attention-based sequence-to-sequence system that also implements the latest state-of-the-art sequence-to-sequence techniques.

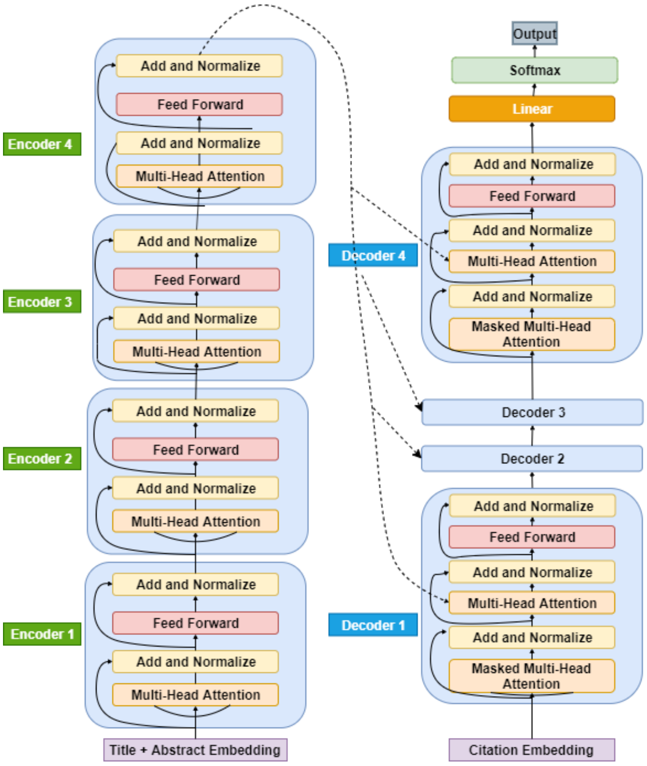


Fig. 5 The Transformer model architecture: encoder to the left and decoder to the right.

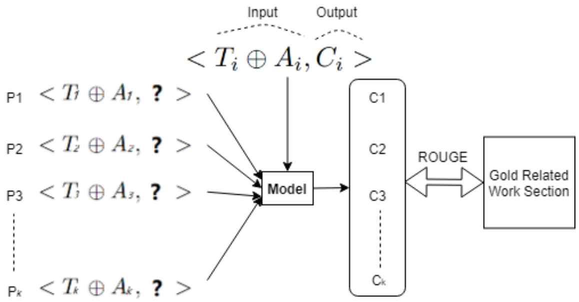

Input
Output
$< T_i \oplus A_i, C_i >$
P1 $< T_1 \oplus A_1,$? $>$
P2 $< T_2 \oplus A_2,$? $>$
P3 $< T_3 \oplus A_3,$? $>$
Pk $< T_k \oplus A_k,$? $>$
Model
C1
C2
C3
Ck
ROUGE
Gold Related Work Section

Fig. 6 Generation of related work sections from a set of papers ($P_1 ... P_n$) and evaluation. Model represents any of the sentence extraction/generation systems tested in this work. Output citation sentences (C_i) are concatenated and compared to a gold standard related work section

.

5 Experiments

In order to compare our approach, we implemented several baselines over the RWSData. Alongside, we ran several experiments to generate abstractive summaries for each cluster i.e. a related work section for a target paper.

5.1 Baselines

For our experiments we implemented several extractive summarization baselines. A set of simple baselines is based on the observations arising from the analysis of citation sentences and scientific abstracts on the use of titles and abstracts (Jaidka et al., 2013; Saggion, 1999).

The *title* baseline is to use the title of each cited article as citation sentences. The *abstract first* baseline uses as citation sentences the first sentence of the abstract of the cited articles while the *abstract last* baseline uses the last sentence.

The second set of baselines is composed of available systems that use well-established extractive techniques. All summarizers are given as input the title and abstract of a cited documents from which a single summary sentence is obtained. They are as follows:

- *MEAD* (Radev et al., 2004) is a well-known extractive document summarizer which generates summaries using centroids alongside other features such as the position of the sentence and the length. We configured MEAD to select one sentence from each cited paper in order to generate the related work section.

- *TextRank* (Mihalcea and Tarau, 2004) and *LexRank* (Erkan and Radev, 2004) are both extractive and unsupervised graph-based text summarization systems which create sentence graphs in order to compute centrality values for each sentence. Both algorithms have similar underlying methods to compute centrality which are based on the PageRank ranking algorithm. They differ in how links are weighted in the document graph.
- *SUMMA* (Saggion, 2008) is a Java implementation of several sentence scoring functions. We use the implementation of the centroid scoring functionality to select the most central sentence in a document.
- *SEQ*^{3} Baziotis et al. (2019) the unsupervised abstractive model named *SEQ*^{3} which used a sequence-to-sequence-to-sequence autoencoder was used as a recently proposed non-extractive technique.

5.2 Extracting Sentences with a Convolutional Neural Network

This system, which is based on a neural network architecture which achieved state of the art performance in the Sci-Summ 2018 Challenge (Abura'ed et al., 2018; Mayr et al., 2019), takes advantage of the potential of convolutions to abstract higher level features from sentences in order to learn its relevance in a specific document (Abura'ed et al., 2017, 2018). This relevance is based on the relationship between a set of features extracted and computed for each sentence and the scoring function. The system assigns a score between 0 (not relevant) and 1 (highly relevant).

5.2.1 Extraction of Sentence Features:

The set of sentence features is organized into two inputs to feed the system. First, we transformed each word from a sentence into a vector by looking up word embeddings. In this scenario, we used two pre-trained word embeddings, which were concatenated: the Google News embeddings[11] (three million words in 300 dimensional vectors trained using word2vec (Mikolov et al., 2013a) over a news text corpus of 100 billion words) and the Association for Computational Linguistics (ACL) Anthology Reference Corpus embeddings (Liu, 2017) (300 dimensional vector trained over a corpus of ACL papers (Bird, 2008)). This embedding matrix representing the words contained in a sentence is introduced in the system as input.

In addition to word embeddings, for each sentence we extracted, using SUMMA (Saggion, 2008; Abura'ed et al., 2018), features in order to provide information about its context in the document:

- Sentence Document Similarity: the cosine similarity of a sentence vector to the article centroid.

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

- Title Sentence Similarity: the cosine similarity of a sentence vector to the vector of the first sentence, that is, the title of the RP.
- TextRank Normalized: a sentence vector is computed to obtain a normalized score using the TextRank algorithm (Mihalcea and Tarau, 2004).
- Position: a score representing the position of the sentence in the article.
- Normalized Cue-phrase: the total number of cue-words in the sentence divided by the total number of cue-words in the article based on (Teufel and Moens, 2002) formulaic expressions.
- Term Frequency: 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 document.
- Rhetorical Class Probability: the probability that the sentence belongs to each of five rhetorical categories – background, outcome, approach, challenge, and future work (five features, one per each rhetorical category) according to the scientific document analyser Dr Inventor (Ronzano and Saggion, 2015).

To calculate the similarities and TextRank Normalized features, we computed three different vectors based on the sentence representations. A vector similarity is the result of comparing two vectors of the same type using the cosine distance function. From the previous input, we also used the Google and ACL pre-trained word embeddings to generate two sentence vectors by calculating the centroid (or average) of the words vectors contained in a sentence. The third vector is based on a SUMMA word vector (Saggion, 2008), which is computed from the tf*idf of each word.

Finally, the context features are also introduced in the system (as a second input) within a sequential window including the context features of the 3 previous and 3 following sentences.

5.2.2 Scoring Functions:

The aim of the system is to learn a scoring function in order to select the most relevant sentences from a document (title + abstract). In other words, the system learns the relation between both set of features (word embeddings and context features) and a score, learning a regression task.

In this work three scoring functions are defined related to the three sentence vectors (SUMMA, Google and ACL), which are basically based on the similarity between sentences in the document (title + abstract) and the gold citation sentence.

5.2.3 Convolutional Model:

The network independently decodes each input (word embeddings and context features) by convolutions to abstract higher level features. Each convolution applies a filter to produce a new feature, which is included in the resulting feature map. The convolution can be replicated with different windows with multiple filters giving multiple feature maps.

Next, a max-pooling layer selects the most relevant feature from each feature map. Relevant features are concatenated together in a single feature vector. In order to prevent over-fitting, after max-pooling layer we applied dropout regularization over the single feature vector (Hinton et al., 2012).

At this point, both single feature vectors generated by each input are also concatenated and the resulting vector is passed to two subsequent fully-connected layers. The fully-connected layers scale a large amount of features from the previous vector to a single output value, in order to learn the regression task. We also rescale the weights whose l2-norms exceed a hyperparameter as in (Kim, 2014) and (Nguyen and Grishman, 2015).

5.3 Sequence-to-Sequence Approach

We feed our training sequences (see Section 3) to the model and use the validation data to tune the hyper parameters and keep the learning rate in check during training. We have used 15,000 pairs for training, 1,514 pairs for development and 219 pairs for testing.

The final model is fed with the set of reference papers (titles and abstracts) in the testing dataset generating a citation context for each reference paper (see Figure 6). Finally, we group the generated set of citations context together to form the final related work section. We ran all our experiments on both the Title and Abstract as described at section 3 and the filtered version of the data.

5.3.1 Training

For our abstractive sequence-to-sequence approach we generated several models while training the data. We ran two separate encoder-decoder architectures i.e. Transformer and BRNN as mentioned at section 4 with 4 recurrent layers for the transformer architecture and one layer of Bi-Directional RNN. We set the hidden size of the recurrent unit to 512 and used ADAM (Kingma and Ba, 2014) and AdaGrad (Duchi et al., 2011) optimizers respectively.
We set the system to share the same weight vectors for shared vocabulary between the encoder and decoder and we add a sinusoidal position encoding to each vocabulary. This option drastically decreases the number of parameters a model has to learn.

To further represent the sentences we not only rely on the internal representation of words by the OpenNMT-py (Klein et al., 2017) tool, but we also use word-based and character-based word2vec pre-trained models. These models will provide some insight of how changing the representation of the input could affect the results, for that reason we use GoogleNews (Mikolov

et al., 2013b) (word-based) and FastText (Mikolov et al., 2018) (character-based) pre-trained models to run additional experiments for both the filtered and unfiltered data.
Regarding batches and normalization, there are two types of batches; sentence based and token based. Sentence based batching sets the batch size based on the number of instances (sentences), while token based batching is also known as dynamic batching due to the fact that a batch is created based on a specific number of tokens. The motivation behind dynamic batching is to avoid any memory problems for sentences that are considered long, it is usually used with greedy algorithms such as the Transformer's multi-head attention technique. For our experiments we batch and normalize based on dynamic batching of size 4,096 tokens for the Transformer architecture. As for the BRNN we batch and normalize based on sentence batching of size 16. We set the network to compute gradients and update the parameters after each set of batches. Moreover, we initialized with Xavier uniform (Glorot and Bengio, 2010) and used 0.2 dropout (Srivastava et al., 2014) mechanism to prevent over-fitting.

We set the network to save models over time, every K steps a model is saved and tested against the validation data generating a total of ten models. See Figure 7 which highlights the accuracy of the network at each check point (Trans is a short for Transformer). The figure shows the accuracy over the training and validation steps for the BRNN and Transformer models over the filtered (denoted as F) and unfiltered data. The BRNN models tend to have a slower and more consistent training accuracy improvements over the transformer models, the slowest learning process were recorded over the filtered data. As for validation accuracy the transformer models are more consistent and stable over the validation data. Finally the validation accuracy of both models have a higher accuracy over the filtered data.

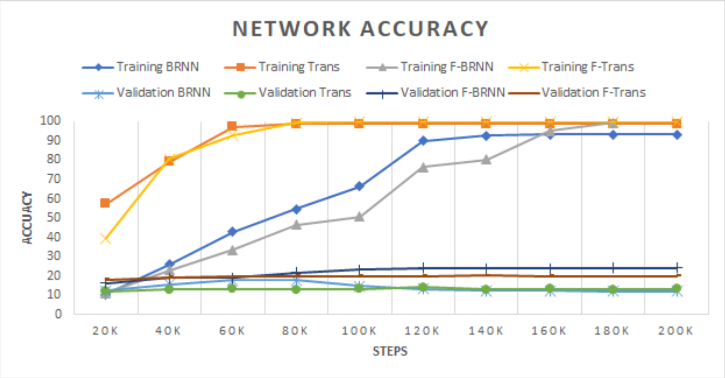


Fig. 7 The neural network accuracy over training and validation data over time

Finally, we used a learning rate decay managed under the ”noam” scheme (Goyal et al., 2017) (linear warm-up for a given number of steps followed by exponential decay of the learning rate).

5.3.2 Testing

We ran the testing sequences over the models generated at each check point. The network reported the perplexity scores (Jelinek et al., 1977) at each check point (See Figure 8) which shows that the Transformer models has less perplexity measures than the BRNN. The generated sentences from our system varied between readable sentences and sentences that were not acceptable, but shared common words with the title and abstract of the cited scientific paper. An example of a good generated citation for a paper in the test set [12] is shown in Figure 9.

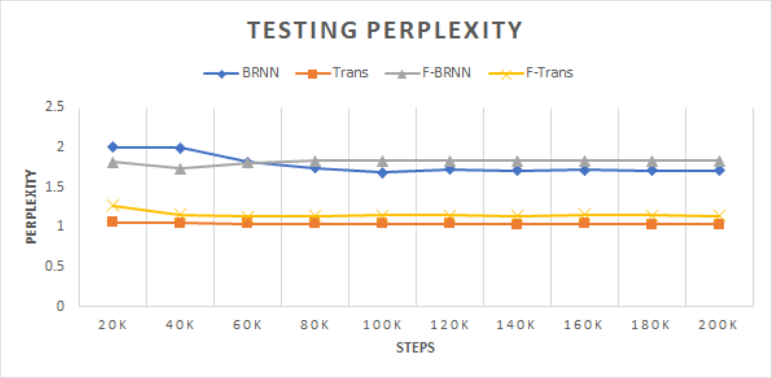


Fig. 8 Perplexity of generated strings at different training points.

An example of a bad generated citation for a testing paper [13] in shown in Figure 10. Even though the generated citation is not very readable due to the inclusion of several “main“ verbs without proper syntactic structure, some relevant keywords have been selected.

Figure 11 shows the entire pipeline of our experiments. We experimented on *title and abstract* of scientific papers and we also applied a filter based on Teufel's (Teufel et al., 2000) gazetteers producing a *title + filtered abstract*. As for the representation of the sentences, we used the internal representation by OpenNMT-py, word-based Word2Vec pre-trained model (i.e. GoogleNews) and character-based word2vec pre-trained model (i.e. FastText). The input

[12] Cited paper: Turney, Peter D. ”Measuring semantic similarity by latent relational analysis.” arXiv preprint cs/0508053 (2005).

[13] Cited paper: Ibrahim, Ali, Boris Katz, and Jimmy Lin. ”Extracting structural paraphrases from aligned monolingual corpora.” Proceedings of the second international workshop on Paraphrasing-Volume 16. Association for Computational Linguistics, 2003.

title	Measuring Semantic Similarity by **Latent Relational Analysis**
abstract	this paper introduces **latent relational analysis** (lra), a **method** for measuring semantic similarity. this paper **describes** ... classifying semantic relations in **noun** modifier expressions. this paper has introduced a new **method** for calculating **relational** similarity, latent relational analysis. just as attributional similarity measures have proven to have many practical uses, ...
gen. cit.	<CITE>describes a **method** (**latent relational analysis**) that extracts subsequence patterns for **noun** pairs from a large corpus, using query expansion to increase the recall of the search and feature selection and dimensionality reduction to reduce the complexity of the features.

Fig. 9 Example of a scientific article (title ⊕ abstract) and a grammatically correct generated citation sentence with considerable "matching" content.

title	Extracting Structural **Paraphrases** From Aligned **Monolingual** Corpora.
abstract	we present an **approach** for automatically learning **paraphrases** from aligned **monolingual** corpora. we present an **approach** for automatically learning **paraphrases** ... our algorithm works by generalizing the syntactic **paths** between corresponding anchors in aligned **sentence** pairs... we also describe a novel **information** retrieval system under development that is designed to take advantage of structural **paraphrases**.
gen. cit.	<CITE >proposed a **information** based **approach** to select **monolingual paraphrases** of a **paraphrases** in the **sentence paths** of a **sentence** paths to reduce the **monolingual** rules of **paraphrases** and penn variations to be identified.

Fig. 10 Example of a scientific article (title ⊕ abstract) and an incoherent generated citation sentence.

Table 1 Extractive Baseline Systems VS Abstractive Sequence-to-Sequence System. ROUGE-1 and ROUGE-2 Metrics.

	ROUGE-1			**ROUGE-2**		
SYSTEM	**R**	**P**	**F**	**R**	**P**	**F**
Titles	0.074	**0.375**	0.119	0.013	**0.072**	0.022
AbsFS	0.126	0.272	0.155	0.019	0.041	0.023
AbsLS	0.114	0.263	0.150	0.013	0.035	0.018
SUMMA	0.130	0.236	0.158	0.019	0.026	0.020
MEAD	**0.247**	0.215	0.219	0.067	0.042	0.048
LexRank	0.162	0.306	0.194	0.029	0.044	0.032
TexRank	0.211	0.232	0.207	0.043	0.038	0.038
SEQ^3	0.045	0.140	0.066	0.0004	0.002	0.0007
CNN_{SUMMA}	0.163	0.262	0.187	0.030	0.047	0.034
CNN_{Google}	0.191	0.261	0.207	0.034	0.0413	0.034
CNN_{ACL}	0.176	0.246	0.195	0.035	0.041	0.035
$Transformer_{CB}$	0.216	0.237	0.215	**0.072**	0.063	0.063
$BRNN_{CB}$	0.189	0.293	0.219	0.054	0.070	0.058
$Transformer_{WB}$	0.221	0.248	0.222	0.070	0.062	0.062
$BRNN_{WB}$	0.179	0.266	0.204	0.044	0.055	0.046
Transformer	0.192	0.255	0.219	0.066	0.071	0.069
BRNN	0.223	0.238	**0.230**	0.069	**0.072**	**0.070**

Table 2 Extractive Baseline Systems VS Abstractive Sequence-to-Sequence System Over the Filtered Data. ROUGE-1 and ROUGE-2 metrics

	ROUGE-1			**ROUGE-2**		
SYSTEM	**R**	**P**	**F**	**R**	**P**	**F**
Titles	0.074	**0.375**	0.119	0.013	0.072	0.022
AbsFS	0.118	0.271	0.150	0.019	0.043	0.024
AbsLS	0.115	0.265	0.146	0.014	0.036	0.019
SUMMA	0.142	0.288	0.180	0.022	0.040	0.027
MEAD	0.216	0.239	0.203	0.038	0.037	0.034
LexRank	0.138	0.292	0.172	0.024	0.038	0.028
TexRank	0.222	0.236	0.210	0.041	0.036	0.035
SEQ^3	0.068	0.158	0.091	0.003	0.006	0.004
CNN_{SUMMA}	0.118	0.250	0.146	0.018	0.038	0.023
CNN_{Google}	0.182	0.234	0.187	0.035	0.038	0.033
CNN_{ACL}	0.187	0.239	0.193	0.037	0.042	0.037
$Transformer_{CB}$	0.276	0.267	0.271	0.120	0.092	0.104
$BRNN_{CB}$	**0.286**	0.314	0.299	**0.122**	**0.108**	**0.115**
$Transformer_{WB}$	0.274	0.276	0.275	0.118	0.092	0.103
$BRNN_{WB}$	0.284	0.317	**0.300**	0.120	0.107	0.113
Transformer	0.261	0.251	0.256	0.116	0.088	0.100
BRNN	0.281	0.298	0.289	0.117	0.100	0.108

source is fed to the pointer generator architecture (BRNN or Transformer) which generates a summary (i.e. citation context) based on the presentation of sentences.

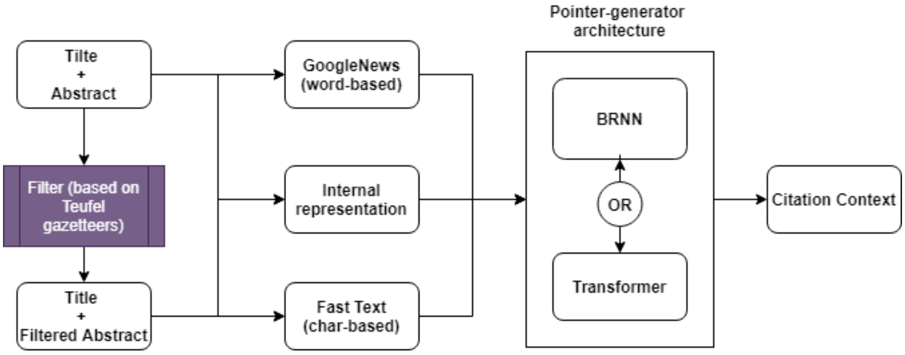


Fig. 11 An outline of the performed experiments showing the different scenarios we used over our approach.

6 Results

In this section we compare our abstractive sequence-to-sequence approaches with the baselines. We used several ROUGE metrics (Lin, 2004) to automatically evaluate all the systems. The metrics used from ROUGE are: ROUGE-L: which uses the *Longest Common Subsequence (LCS)* evaluating the structural similarity between two summaries therefore paying attention to syntax; ROUGE-1: which checks the overlap of each word between the automated summary and the gold standard paying attention to word content; ROUGE-2: similar to ROUGE-1 but at the level of bi-gram overlap; and finally ROUGE-SU4: which considers Skip-bigram plus unigram-based co-occurrence statistics therefore considering long sequences as the basis for evaluation. ROUGE measures combine precision and recall in a harmonic F-measure which is generally used to assess the systems' performance.

The results of ROUGE-1 and ROUGE-2 metrics before filtering the data can be found at Table 1 and over the filtered data at Table 2. ROUGE-L and ROUGE-SU4 results are computed for the sake of completeness and provided in the Appendix in tables 7 and 6 for unfiltered and filtered data respectively.

As can be appreciated from the numbers in Tables 1, 6, 2 and 7 the non-informed extractive baselines which do not perform any analysis of the input (e.g. use of titles or sentences from abstracts) tend to have a high precision but low recall, specially precise is the title. For all ROUGE measures, and disregarding of the status of the input data (filtered/non-filtered), the sequence-to-sequence models obtain the higher scores in terms of F-score (ROUGE-F). For precision and recall variants of ROUGE in the case of non-filtered data, we can observe that MEAD is better at Recall and LexRank at precision, however not achieving the best F-score. This trend is not observed in the filtered data where the sequence-to-sequence models obtain higher results for precision, recall, and F-score (for all ROUGE measures).

In order to eliminate our bias and get a better insight of how reliable our methods are over the filtered and unfiltered data in comparison with the baselines, we have analysed the ROUGE results by running a $t-$test[14] (using the R software and selecting 95% confidence level). We report our analysis on Tables 3 and 4 for the differences when the same approach is trained with different data types (filtered vs. non-filtered). Moreover, for each sequence-to-sequence model we analyze the effect of the embedding condition used (none, word embedding, character embedding), see Table 5. More specifically, Table 3 compares ROUGE-1 means of the different systems under the filtered and non-filtered conditions. We can observe that differences are statistically significant for all sequence-to-sequence models († in the sig. column indicates if a difference was found). Table 4 compares ROUGE-2 results showing similar findings, the filtered condition offers clear advantages. Table 5 compares ROUGE (1 and 2) means for the BRNN and Transformer approaches (under

[14] Normality of the data was verified with a Kolmogorov-Smirnov test of normality.

different embedding conditions). Differences are statistically significant for 9 out of 12 conditions († in the sig. column indicates if a difference was found) indicating that BRNN is superior in most conditions. Besides, statistical tests (not shown in the tables) comparing BRNN with different embedding conditions indicate that only in two cases of non-filtered data: (i) none/word embedding ($p < 0.06$) and (i) character/word embedding ($p < 0,007$), differences exist. When comparing the embedding condition for Transformer (not shown in tables), only one difference is detected: the none/character embedding with filtered data ($p < 0.06$).

Table 3 Comparison of *filtered* vs. *non-filtered* ROUGE-1 results with two-tailed t-test(20). Mean, standard deviation (sd), and p-values (sig.) are reported.

System	Filtered		Non-filtered		
	mean	sd	mean	sd	sig.
BRNN	0.28	0.002	0.23	0.0008	$1 * 10^{-4\dagger}$
BRNN$_{CB}$	0.29	0.002	0.21	0.0009	$9.86 * 10^{-7\dagger}$
BRNN$_{WB}$	0.30	0.002	0.20	0.001	$2 * 10^{-8\dagger}$
Transf	0.25	0.003	0.21	0.0005	$0.01^{\dagger}$
Transf$_{CB}$	0.27	0.001	0.21	0.001	$2.39 * 10^{-6\dagger}$
Transf$_{WB}$	0.27	0.001	0.22	0.001	$2 * 10^{-6\dagger}$
CNN$_{SUMMA}$	0.14	0.001	0.18	0.003	$8 * 10^{-4\dagger}$
CNN$_{Google}$	0.18	0.001	0.20	0.003	0.14
CNN$_{ACL}$	0.19	0.001	0.19	0.002	0.86
SUMMA	0.18	0.001	0.15	0.001	$0.01^{\dagger}$
MEAD	0.20	0.004	0.21	0.003	0.2
LexRank	0.17	0.001	0.19	0.003	0.09
TextRank	0.21	0.002	0.20	0.002	0.78

Table 4 Comparison of *filtered* vs. *non-filtered* ROUGE-2 results with two-tailed t-test(20). Mean, standard deviation (sd), and p-values (sig.) are reported.

System	Filtered		Non-filtered		
	mean	sd	mean	sd	sig.
BRNN	0.10	0.002	0.07	0.0003	$8 * 10^{-2\dagger}$
BRNN$_{CB}$	0.11	0.002	0.058	0.0003	$1.6 * 10^{-4\dagger}$
BRNN$_{WB}$	0.11	0.002	0.047	0.002	$2.45 * 10^{-6\dagger}$
Transf	0.10	0.002	0.069	0.0002	$1.6 * 10^{-2}\dagger$
Transf$_{CB}$	0.10	0.0018	0.063	0.0002	$1 * 10^{-3\dagger}$
Transf$_{WB}$	0.10	0.002	0.62	0.0001	$7 * 10^{-4\dagger}$
CNN$_{SUMMA}$	0.023	0.0001	0.034	0.0005	$0.02^{\dagger}$
CNN$_{Google}$	0.035	0.0003	0.034	0.0005	0.83
CNN$_{ACL}$	0.037	0.0005	0.035	0.0007	0.77
SUMMA	0.027	0.00027	0.019	0.0001	0.09
MEAD	0.034	0.00098	0.049	0.00076	$2 * 10^{-3\dagger}$
LexRank	0.028	0.0002	0.032	0.0002	0.39
TextRank	0.035	0.0004	0.038	0.00035	0.47

Table 5 Comparison of ROUGE scores in BRNN and Transformer systems under different embedding conditions using two-tailed *t*-test(20). Mean, standard deviation (sd), and p-values (sig.) are reported.

Embedding	**BRNN**		**Transf**		**sig.**
		ROUGE-1 Filtered			
	mean	sd	mean	sd	
None	0.28	0.002	0.25	0.003	$1.09 * 10^{-6}$†
Word	0.30	0.002	0.27	0.001	0.003†
Character	0.29	0.002	0.27	0.001	0.0002†
		ROUGE-1 Non-filtered			
	mean	sd	mean	sd	
None	0.23	0.0008	0.21	0.0005	0.015†
Word	0.22	0.001	0.20	0.001	0.008†
Character	0.21	0.0009	0.21	0.0009	0.58
		ROUGE-2 Filtered			
	mean	sd	mean	sd	
None	0.108	0.002	0.100	0.002	0.001†
Word	0.11	0.002	0.10	0.001	$8.5 * 10^{-5}$†
Character	0.11	0.002	0.10	0.001	0.0029†
		ROUGE-2 Non-filtered			
	mean	sd	mean	sd	
None	0.07	0.0003	0.069	0.0001	0.50
Word	0.06	0.0001	0.046	0.0002	0.0001†
Character	0.06	0.0002	0.058	0.0003	0.11

7 Limitations

Certain limitations apply to abstractive summarization methods in which the generated text could be repetitive for certain phrases that appears often in the training data (e.g. stop words), such repetition could affect the comprehensibility of the text. Using a huge dataset as training could reduce the repetition also some post-processing steps could be applied. We have utilized OpenNMT-py to prevent the model from repeating trigrams in the same sentence (i.e. block_ngram_repeat argument), which could help addressing this problem. Example of an incoherent sentence in Figure 10 shows that the syntactic structure of the outcome text should be improved. Denoising is one of the promising techniques to tackle this issue (Artetxe et al., 2018). It consists in reordering the input sequence and reconstructing the original word order that makes the model learn how to compose words to result in correct syntactic transformation. It is relevant to our task since there are cases when the change of positions of words in a citation sentence and a corresponding change in the syntactic structure are required to compose a meaningful summary (cf., the upper-right text in Figure 3). We are going to try this technique in the future taking into account that according to the recent works in denoising (Surya et al., 2019) for complex syntactic operations such as sentence splitting, rephrasing, and paraphrasing, some explicit mechanisms should be employed in addition.

Although our work is related to a number of scientific summarization approaches, the work most similar to ours is (Hu and Wan, 2014) who made available the dataset of related work sections used in our evaluation. Their approach however can not be compared directly with ours due to several facts but most importantly: (i) their software is not available to run and (ii) their paper does not indicate which part of the corpus was used for evaluation, leaving reproducible research of their approach difficult to achieve[15]. In spite of this limitation, we argue that the complete comparison of approaches we have carried out here provides a solid picture into the use of sequence-to-sequence approaches for this specific summarization task.

8 Conclusion

Being an essential part of every scientific article, related work sections or literature reviews pose important challenges for natural language processing in the context of the scientific text. Here we have been concerned with the generation of "descriptive" related work section given a set of scientific papers to summarize. Based on previous research, which indicate that related work sections usually include elements from titles and abstracts of the cited papers, we have reduced the complexity of the task considering as input to our generation process only those parts of the scientific articles. Since it has also been shown that related work sections exhibit cut-and-paste summarization strategies we have investigated a sequence-to-sequence approach in order to automatically generate citation sentences for each paper to cite. Our sequence-to-sequence approach makes use of a novel dataset which we make available to the research community for further research. We additionally have presented a comparison between our abstractive approach against a set of extractive methods and evaluated them based on a gold standard dataset using content-based metrics. Our results indicate that our approach outperforms the simple as well as the informed baselines and competitive neural network approaches. There are many avenues for continuing this research such as considering a broader approach to the generation of citations which will jointly take advantage of existing citations as well as paper content to generate more informed citations. Our approach for now just uses titles and abstracts of scientific papers as a source and the citation context as the target because it is cost effective to access them, MAG API facilitates having such information by indexing it directly. However, we will investigate the cost of adding more sentences from the scientific papers directly and the value of extending the context of the source and the target. Another important direction of research is to investigate how to generate sentences which combine in a given sentence information from multiple papers. Future work should also examine the role of discourse

[15] We have attempted to contact in several occasions the authors without receiving any answers.

and how to connect different citation sentences to produce a cohesive and coherent piece of text. In this sense, a subject which would be interesting to address is that of generating integrative reviews which compare, contrast, and provide judgments on papers, putting them in context. However, we see this as a very challenging task in that it would require specific knowledge besides linguistic one to understand which aspects should be compared and in which way, a topic which would be difficult to address with the techniques we have presented here.

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 Special thanks to ALI TAQI for designing the pointer generator architecture figure.

References

2008 — 2019. Grobid. `https://github.com/kermitt2/grobid`.

Ahmed AbuRa'ed, Horacio Saggion, and Luis Chiruzzo. 2020. A multi-level annotated corpus of scientific papers for scientific document summarization and cross-document relation discovery. In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC 2020).*

Ahmed Abura'ed, Alex Bravo, Luis Chiruzzo, and Horacio Saggion. 2018. Lastus/taln+ inco@ cl-scisumm 2018-using regression and convolutions for cross-document semantic linking and summarization of scholarly literature. In *Proceedings of the 3nd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL2018). Ann Arbor, Michigan (July 2018).*

Ahmed Abura'ed, Luis Chiruzzo, Horacio Saggion, Pablo Accuosto, and Àlex Bravo Serrano. 2017. Lastus/taln@ clscisumm-17: cross-document sentence matching and scientific text summarization systems. In *Proc. of the 2nd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL2017). Tokyo, Japan (August 2017).*

Nitin Agarwal, Kiran Gvr, Ravi Shankar Reddy, and Carolyn Penstein Rosé. 2011. Towards multi-document summarization of scientific articles: making interesting comparisons with scisumm. In *Proceedings of the Workshop on Automatic Summarization for Different Genres, Media, and Languages*, pages 8–15. Association for Computational Linguistics.

Nouf Ibrahim Altmami and Mohamed El Bachir Menai. 2018. Semantic graph based automatic summarization of multiple related work sections of scientific articles. In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, pages 255–259. Springer.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018.*

Christos Baziotis, Ion Androutsopoulos, Ioannis Konstas, and Alexandros Potamianos. 2019. Seqˆ 3: Differentiable sequence-to-sequence-to-sequence autoencoder for unsupervised abstractive sentence compression. *arXiv preprint arXiv:1904.03651.*

Florian Beil, Martin Ester, and Xiaowei Xu. 2002. Frequent term-based text clustering. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 436–442. ACM.

Steven Bird. 2008. The ACL anthology reference corpus: A reference dataset for bibliographic research in computational linguistics.

Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2019. Unsupervised multi-document opinion summarization as copycat-review generation. *arXiv preprint arXiv:1911.02247.*

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078.*

Eric Chu and Peter J Liu. 2018. Meansum: a neural model for unsupervised multi-document abstractive summarization. *arXiv preprint arXiv:1810.05739.*

Alexandru Constantin, Steve Pettifer, and Andrei Voronkov. 2013. Pdfx: fully-automated pdf-to-xml conversion of scientific literature. In *Proceedings of the 2013 ACM symposium on Document engineering*, pages 177–180. ACM.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Brigitte Endres-Niggemeyer, Elisabeth Elisabeth Maier, and Alexander Alexander Sigel. 1995. How to implement a naturalistic model of abstracting: Four core working steps of an expert abstractor. *Information Processing and Management*, 31(5):631 – 674.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Daniel Ferrés, Horacio Saggion, Francesco Ronzano, and Àlex Bravo. 2018. Pdfdigest: an adaptable layout-aware pdf-to-xml textual content extractor for scientific articles. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018).*

Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. 2018. Bottom-up abstractive summarization. *arXiv preprint arXiv:1808.10792.*

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. 2017. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677.*

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580.*

Cong Duy Vu Hoang and Min-Yen Kan. 2010. Towards automated related work summarization. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 427–435. Association for Computational Linguistics.

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM.

Yue Hu and Xiaojun Wan. 2014. Automatic generation of related work sections in scientific papers: An optimization approach. In *EMNLP*, pages 1624–1633.

Kokil Jaidka, Muthu Kumar Chandrasekaran, Rahul Jha, Christopher Jones, Min-Yen Kan, Ankur Khanna, Diego Mollá-Aliod, Dragomir R. Radev, Francesco Ronzano, Horacio Saggion, and Wee Kim Wee. 2014. The computational linguistics summarization pilot task. In *Proceedings of TAC 2014.*

Kokil Jaidka, Christopher Khoo, and Jin-Cheon Na. 2013. Deconstructing human literature reviews–a framework for multi-document summarization. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 125–135.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2014. On using very large target vocabulary for neural machine translation. *arXiv preprint arXiv:1412.2007.*

Sébastien Jean, Orhan Firat, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. Montreal neural machine translation systems for wmt'15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 134–140.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63.

Rahul Jha, Amjad Abu-Jbara, and Dragomir R Radev. 2013. A system for summarizing scientific topics starting from keywords. In *ACL (2)*, pages 572–577.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882.*

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810.*

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain.

Haixia Liu. 2017. Sentiment analysis of citations using word2vec. *CoRR*, abs/1704.00177.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 76–79.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025.*

Minh-Thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. 2014. Addressing the rare word problem in neural machine translation. *arXiv preprint arXiv:1410.8206.*

Lauren Maggio, Justin Sewell, and Anthony Artino. 2016. The literature review: A foundation for high-quality medical education research. *Journal of Graduate Medical Education*, 8:297–303.

Diana Maynard, Valentin Tablan, Hamish Cunningham, Cristian Ursu, Horacio Saggion, Kalina Bontcheva, and Yorick Wilks. 2002. Architectural elements of language engineering robustness. *Natural Language Engineering*, 8(2-3):257–274.

Philipp Mayr, Muthu Kumar Chandrasekaran, and Kokil Jaidka. 2019. Report on the 3rd joint workshop on bibliometric-enhanced information retrieval and natural language processing for digital libraries (birndl 2018). In *ACM SIGIR Forum*, volume 52, pages 105–110. ACM.

Sean M McNee, Istvan Albert, Dan Cosley, Prateep Gopalkrishnan, Shyong K Lam, Al Mamunur Rashid, Joseph A Konstan, and John Riedl. 2002. On the recommending of citations for research papers. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pages 116–125. ACM.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing.*

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *ICLR Workshop.*

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018).*

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Saif Mohammad, Bonnie Dorr, Melissa Egan, Ahmed Hassan, Pradeep Muthukrishan, Vahed Qazvinian, Dragomir Radev, and David Zajic. 2009. 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.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48.

Michael P. Oakes and Chris. D. Paice. 1999. The automatic generation of templates for automatic abstracting. In *Proceedings of the 21st Annual BCS-IRSG Conference on Information Retrieval Research*, IRSG'99, pages 11–11, Swindon, UK. BCS Learning & Development Ltd.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Vahed Qazvinian, Dragomir R Radev, Saif Mohammad, Bonnie J Dorr, David M Zajic, Michael Whidby, and Taesun Moon. 2013. Generating extractive summaries of scientific paradigms. *J. Artif. Intell. Res.(JAIR)*, 46:165–201.

Dragomir R Radev, Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda Celebi, Stanko Dimitrov, Elliott Drabek, Ali Hakim, Wai Lam, Danyu Liu, et al. 2004. Mead-a platform for multidocument multilingual text summarization. In *LREC*.

Francesco Ronzano and Horacio Saggion. 2015. Dr. Inventor Framework: Extracting structured information from scientific publications. In *International Conference on Discovery Science*, pages 209–220. Springer.

Francesco Ronzano and Horacio Saggion. 2016. An empirical assessment of citation information in scientific summarization. In *Natural Language Processing and Information Systems - 21st International Conference on Applications of Natural Language to Information Systems, NLDB 2016, Salford, UK, June 22-24, 2016, Proceedings*, pages 318–325.

Jennifer Rowley and Frances Slack. 2004. Conducting a literature review. *Management Research News*, 27.

Christopher S. G. Khoo, Jin-Cheon Na, and Kokil Jaidka. 2011. Analysis of the macro-level discourse structure of literature reviews. *Online Information Review*, 35.

Horacio Saggion. 1999. Using linguistic knowledge in automatic abstracting. In *27th Annual Meeting of the Association for Computational Linguistics, University of Maryland, College Park, Maryland, USA, 20-26 June 1999.*

Horacio Saggion. 2008. SUMMA: A robust and adaptable summarization tool. *Traitement Automatique des Langues*, 49(2):103–125.

Horacio Saggion. 2011. Learning predicate insertion rules for document abstracting. In *Computational Linguistics and Intelligent Text Processing - 12th International Conference, CICLing 2011, Tokyo, Japan, February 20-26, 2011. Proceedings, Part II*, pages 301–312.

Horacio Saggion and Guy Lapalme. 2002. Generating indicative-informative summaries with sumum. *Computational linguistics*, 28(4):497–526.

Horacio Saggion and Thierry Poibeau. 2013. Automatic text summarization: Past, present and future. In *Multi-source, Multilingual Information Extraction and Summarization*, pages 3–21.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-june Paul Hsu, and Kuansan Wang. 2015. An overview of microsoft academic service (mas) and applications. In *Proceedings of the 24th international conference on world wide web*, pages 243–246. ACM.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Sai Surya, Abhijit Mishra, Anirban Laha, Parag Jain, and Karthik Sankaranarayanan. 2019. Unsupervised neural text simplification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2058–2068.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Jie Tang. 2016. Aminer: Toward understanding big scholar data. In *WSDM*.

Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 990–998. ACM.

Simone Teufel and Marc Moens. 2002. Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational linguistics*, 28(4):409–445.

Simone Teufel et al. 2000. *Argumentative zoning: Information extraction from scientific text*. Ph.D. thesis, Citeseer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015a. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015b. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

Hoang Cong Duy Vu. 2010. *Towards automated related work summarization.* Ph.D. thesis.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R Fabbri Irene Li Dan, and Friedman Dragomir R Radev. 2019a. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander Fabbri, Irene Li, Dan Friedman, and Dragomir Radev. 2019b. ScisummNet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *Proceedings of AAAI 2019*.

Xingxing Zhang, Mirella Lapata, Furu Wei, and Ming Zhou. 2018. Neural latent extractive document summarization. *arXiv preprint arXiv:1808.07187*.

A ROUGE-L and ROUGE-SU4 Metrics Results

We present here the results of our experiments over the filtered and unfiltered data.

Table 6 Extractive Baseline Systems VS Abstractive Sequence-to-Sequence System.ROUGE-L and ROUGE-SU4 Metrics

	ROUGE-L			**ROUGE-SU4**		
SYSTEM	**R**	**P**	**F**	**R**	**P**	**F**
Titles	0.087	0.363	0.134	0.029	**0.147**	0.046
AbsFS	0.149	0.260	0.174	0.051	0.082	0.056
AbsLS	0.127	0.221	0.151	0.045	0.079	0.054
SUMMA	0.129	0.186	0.146	0.052	0.059	0.052
MEAD	**0.209**	0.178	0.179	**0.130**	0.067	0.082
LexRank	0.161	0.259	0.183	0.067	0.092	0.070
TexRank	0.186	0.194	0.178	0.092	0.067	0.073
SEQ^3	0.043	0.281	0.074	0.016	0.038	0.021
CNN_{SUMMA}	0.170	0.227	0.181	0.070	0.081	0.070
CNN_{Google}	0.201	0.225	0.199	0.081	0.077	0.073
CNN_{ACL}	0.191	0.206	0.189	0.077	0.075	0.071
$Transformer_{CB}$	0.190	0.189	0.179	0.103	0.077	0.084
$BRNN_{CB}$	0.070	**0.365**	0.103	0.090	0.096	0.088
$Transformer_{WB}$	0.198	0.198	0.189	0.105	0.078	0.085
$BRNN_{WB}$	0.077	0.358	0.118	0.080	0.083	0.077
Transformer	0.166	0.228	0.192	0.098	0.089	0.093
BRNN	0.192	0.213	**0.202**	0.110	0.091	**0.099**

Table 7 Extractive Baseline Systems VS Abstractive Sequence-to-Sequence System Over the Filtered Data. ROUGE-L and ROUGE-SU4 metrics

	ROUGE-L			**ROUGE-SU4**		
SYSTEM	**R**	**P**	**F**	**R**	**P**	**F**
Titles	0.087	0.363	0.134	0.029	0.147	0.046
AbsFS	0.143	0.261	0.171	0.048	0.082	0.055
AbsLS	0.131	0.236	0.157	0.045	0.078	0.052
SUMMA	0.154	0.243	0.178	0.058	0.085	0.066
MEAD	0.179	0.190	0.166	0.093	0.070	0.072
LexRank	0.157	0.264	0.179	0.056	0.092	0.062
TexRank	0.204	0.196	0.187	0.093	0.068	0.073
SEQ^3	0.078	0.205	0.109	0.024	0.042	0.029
CNN_{SUMMA}	0.141	0.230	0.162	0.047	0.075	0.052
CNN_{Google}	0.189	0.197	0.179	0.077	0.069	0.066
CNN_{ACL}	0.191	0.203	0.185	0.082	0.072	0.071
$Transformer_{CB}$	0.231	0.231	0.231	0.155	0.097	0.119
$BRNN_{CB}$	0.117	0.437	0.184	0.163	0.117	0.136
$Transformer_{WB}$	**0.238**	0.235	**0.237**	0.153	0.099	0.120
$BRNN_{WB}$	0.137	0.415	0.206	**0.165**	**0.119**	**0.138**
Transformer	0.225	0.215	0.220	0.145	0.092	0.112
BRNN	0.124	**0.444**	0.193	**0.165**	0.113	0.134