Sequence-Based Extractive Summarisation for Scientific Articles

Daniel Kershaw
d.kershaw@elsevier.com
Elsevier Ltd
London, United Kingdom

Rob Koeling
r.koeling@elsevier.com
Elsevier Ltd
London, United Kingdom

ABSTRACT

This paper presents the results of research on supervised extractive text summarisation for scientific articles. We show that a simple sequential tagging model based only on the text within a document achieves high results against a simple classification model. Improvements can be achieved through additional sentence-level features, though these were minimal. Through further analysis, we show the potential of the sequential model relying on the structure of the document depending on the academic discipline which the document is from.

CCS CONCEPTS

• **Computing methodologies** → **Information extraction**; **Neural networks**.

KEYWORDS

summerisation, neural networks, extractive, corpus, scientific texts

ACM Reference Format:
Daniel Kershaw and Rob Koeling. 2022. Sequence-Based Extractive Summarisation for Scientific Articles. In *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3487553.3524637

1 INTRODUCTION

The rate at which scientific knowledge is created and published is growing continuously [13]. Researchers are spending an increasing amount of time reading and understanding scientific documents. Automated summarisation is seen as a way of serving a more condensed version of an article to a reader in order to aid their reading experience and save time to acquire the necessary knowledge in their field.

Research in automatic summarisation can roughly be separated in abstractive and extractive methods. Traditionally, extractive methods focused on core methods such as unsupervised graph-based methods [6, 16] and supervised model-based approaches [5]. Lately, there have been advances in the use of deep-learning for extractive and abstractive summarisation which resulted in the ability to get close to human quality automated summarisation [17].

This paper shows the results for applying sequential deep learning models to extractive summarisation on a large body of scientific publications. The models are trained and evaluated using author provided document summaries, consisting of a small number of take-away points highlighting the document's contributions.

Our main contributions are:

(1) We introduce the use of a RNN sequential tagger approach for extractive summarisation of scientific documents, which surpasses the baselines through introduction of global context
(2) The addition of hand-engineered features derived from sentences, improving the model, though not significantly
(3) We show that the importance of document structure differs between scientific domains

We show the ability to apply extractive summarisation at scale for scientific documents, where the results are comparable to human performance. The paper is structured as follows: related work in section 2, data sets are introduced in sections 3 with the overarching model explained in section 4. Training of the model along with all model settings are discussed in section 5, followed by the results in section 6 (this includes human evaluation). Finally, section 7 concludes the work with a discussion of the results in the wider context.

2 RELATED WORK

Over the years, we have seen a wide variety of NLP research focussed on scientific articles: citations classification [4], knowledge graph extraction [14], methods identification [15] and citation networks [23] to name a few. Whereas some of these tasks are easy to explore with large open data-sets, others, such as knowledge graph creation, require access to curated annotated data; like those released for the SemEval 2017 and 2018 shared tasks [2].

The field of text summarisation has been furthered through the availability of corpora such as the CNN/DailyMail [22], or social media data-sets such as Reddit [1]. This has lead to developments in abstractive and extractive summarisation models, such as a bidirectional RNN as the base of the model [17]. Kedzie et al. compared several extractive RNN architectures and showed there were limited performance improvements through variations of the model architectures, such as the inclusion of attention [18].

Focusing on the scientific domain Collins et al. showed that one could use extractive summarisation on scientific articles by classifying each sentence as 'summary like' or not. But as each sentence was independently classified, there were issues with the lack of global context. An alternative approach was proposed by Jaidka et al., who used 'Citations'[1] to generate a summary of the document based on what other documents have said about it. This approach results in summaries of what peers actually think of the research, rather than the authors themselves. Though traditionally, methods

[1]This is the contextual sentence around a citation

WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France.

ACM ISBN 978-1-4503-9130-6/22/04…$15.00
https://doi.org/10.1145/3487553.3524637

for summarising scientific articles focus on the patterns within the text with, for example, Teufel and Moens looking at the rhetoric patterns within the text to extract summaries.

In this paper we show that through the application of neural extractive summarisation we can generate short (4 sentence), human readable summaries which match a human baseline on a large corpus of scientific documents from across a variety of scientific disciplines.

3 DATA SET

In this research study we leverage so-called author-provided highlights to train a extractive summarization model. *'Author highlights'* consist of four to six bullet points provided by the author at time of submission, and are designed to be a *'key finding'* summary of the paper, which is more condensed than the abstract. Author highlights are similar to the summaries provided within the CNN/DailyMail data-set [22], which is used extensively in summarisation research. Example author highlights are found in Table 5.

The data-set contains 138,735 documents, this is split into train, test and validation sets. The train and test sets are used during the model training and the validation set is used to report the results presented in sections 6. The documents in this data-set have on average 4 author highlight, with an average of 12 tokens per highlight. This is in comparison to the CNN/DailyMail data-sets which has 4 summary sentences, with a average length of 13 tokens. One notable difference between the two data-sets is that while the CNN/DailyMail documents have on average 22 and 30 sentences respectively, the scientific documents have an average of 128 sentences. Other notable differences are: the more complex language and a higher number of OOV terms used in scientific writing and the different structure in which the articles are written.

3.1 Data Sampling

The training data for the model is generated by selecting sentences from within the documents that are 'similar to' the author provided highlights. Nallapati et al. proposed the use of a greedy sampling method to select the best sub-set of 'similar' sentences. In greedy sampling one sentence is added to the selected set at a time, based on which one increases the overall similarity metric the most. For this work we use `rouge-l-f` as the metric to compare sentences to the set of highlights. Selection of new sentences is stopped once 10 sentences have been selected. Unlike Kedzie et al. and Collins et al., who optimised for `rouge-2-r`, `rouge-l-f` was chosen as it was shown to yield balanced summaries which contain the core topics. This can be seen as a trade-off between recall based metrics, which tend to produce large amounts of noise, and precision based metrics that favour the shortest summaries.

The articles within our data contain a number of features not seen in other similar data-sets. Scientific articles are written in a more formalised structure, with the documents broken down into distinct (sub)sections to aid the reader in understanding the content. Using a gazetteer which maps section titles to a set of high level section types[2], it was shown that 29.9% of sentences were taken from the 'Results' section. This would be in line with the author highlights focusing on the results and findings of the article.

Additionally, author highlights are on average 12 tokens long, compared to an average of 15 tokens for the sentences sampled from the document using `rouge-l-f`. Sampling with `rouge-l-r` results in much longer sentences: an average of 28 tokens. This is evidence that the sentences sampled using `rouge-l-f` are more similar to the author provided highlights.

3.2 Preprocessing

Before the documents are used to train the model they go through a number of preprocessing steps. We normalise or remove non-standard elements typical for scientific documents, like mathematical formulae, chemical compounds and maps. In addition custom XML mark-up is removed, which could contain citation information. This leaves us with the plain text, which is normalised and tokenized using the NLP processing pipeline from JohnSnow Lab.[3]

4 MODEL

We are aiming to select the best subset of sentences, representing the main take-away points (highlights) from an article. As with [10] and [17] we treat this as a sequence tagging problem. Given an article D that contains a set of n ordered sentences $s_0, s_1, ...s_n$, a summary is generated by predicting the label of the sentences $y_1, ..., y_n \in \{0, 1\}^n$, such that $y_i = 1$ means the i^{th} sentence should be included in the summary.

At a high level the proposed models can be broken down into two components, the sentence encoder (see section 4.1) and the sentence extractor (see section 4.2). The sentence encoder processes each individual sentence to form a vector representation, and the sentence extractor then takes all sentence encodings in order to select the best sentences to be included in the highlights. This means that given a sequence of sentence embeddings $h = h_1, h_2, ..., h_n$ the extractor outputs a sequence of predictions (probabilities) in the form of $y = y_1, y_2, ..., y_n$.

Kedzie et al. showed there is limited difference between the results of varying RNN sequence tagging extractive summarisation models, such as [3] which included an attention mechanism, or Nallapati et al. who processed the document twice. For this reason a simple single layered bi-direction RNN was chosen as the base of the summarisation model where the classification was made on the concatenation of the forward and backwards hidden states of each sentence which is then passed through a Multilayer Perceptron (MLP).

4.1 Sentence Encoders

Kedzie et al. proposed three distinct sentence encoders, each representing the sentences in different ways. These are: averaging word embeddings (MEAN), RNN and CNN encoding of the word vectors.

MEAN. For a sentence of n words, the embedding is simply the average over the set of word embeddings $h = \frac{1}{|s|} \sum_{1=1}^{|s|}$

CNN. A number of $1D$ CNN's with varying widths are passed over the word embeddings to produce a sentence embedding. This is similar to [11], though instead of classifying the final layer with

[2]Introduction, Methods, Results, Discussions, Conclusion, Other

[3]https://nlp.johnsnowlabs.com/

	# Documents	Average Labels	Average # Sentence	Avg sentence length
Test	41,756	10.04	162.85	24.07
Train	83,153	10.02	163.62	24.06
Validation	13,826	10.07	163.51	24.13

Table 1: General statistics for the data-sets depending on the domains they represent

a number of dense layers, it is used as an input for the next level of the model.

RNN. As in [5] sentences can be represented as the concatenated final states of a single layered bi-directional RNN. Though instead of using a GRU for the RNN cell an LSTM [7] is used, to replicate the sentence encoding found in [5].

4.1.1 Additional Features. Collins et al. and Narayan et al. demonstrated that both extractive and abstractive summarisation models can be improved with the addition of pre-computed features (side information) in the model. This is achieved by concatenating additional features on to the sentence encodings. For this work, the sentence level features proposed in [5] are used. These are:
Number of numbers - the number of numeric tokens within the sentence.
Sentence Length - the number of tokens within the sentence.
Section Classification - a one-hot encoding of the section class where the sentence comes from. This can either be 'Introduction', 'Related Work', 'Methods', 'Results', 'Discussions', 'Conclusion' or 'Other'. This is done through a simple string match against a gazetteer of section titles.
Title Overlap - the normalised number of words which appear in both the sentence and the title.
Key Phrase Overlap - the number of words which appear in both the key term list and sentence.
Abstract Overlap - the number of words which appear in both the sentence and abstract of the document.

These features are concatenated together into one feature vector per sentence. The feature vector is then passed through a number of dense layers before being concatenated with the sentence embedding. This then makes up the input for the document encoder.

4.2 Sentence Extractors

Kedzie et al. compared a number of sequence based models for extractive summarisation. For simplicity this paper focuses on a simple single layered bi-directional RNN based sequence tagging model. The output of the forward and backwards pass of each cell within the RNN are concatenated and then passed to a MLP layer for final classification. The `soft-max` output of the classifier is then regarded as the probability whether the sentence should be included in the summary or not. This is a discriminatory classifier $p(y_{1:n}|h_{1:n})$. This means that each prediction is made independent of the other predictions by the model. As with one of the sentence encoders proposed in section 4.1 LSTM cells are used within the RNN.

4.2.1 Modification. Section 4.1.1 introduced sentence level features in addition to the sentence embedding. At a document level there are also features which can be used within the sentence extractor. Though instead of concatenating them to an output of the model (as with sentences in section 4.1.1), the hidden states of the LSTM cells within the RNN are initialised with document level features. Initialising the hidden state with user and product attributes was done by Ni et al., Ni et al., Ni and McAuley to generate personal abstractive reviews for products. Instead of using product and user based features, we initialise the LSTM with the ASJC codes, the abstract and the title for the given document.
ASJC - each document is given a series of codes which represent the subject area and disciplines[4] of the journal in which the paper was published. A journal can be associated with multiple All Science Journal Classification (ASJC) codes, thus for a document the final ASJC vector is the normalised sum of the individual ASJC embeddings.
Title - the title is represented by the average of its word embeddings
Abstract - the abstract is represented by the average of its word embeddings

The vectors which represent these three additional features are concatenated together and passed though a fully connected layer before being used to initialise the LSTM within the sentence extractor.

4.3 Base Model

We approach extractive summarisation as a sequence tagging problem. In contrast, Collins et al. [5] classified each individual sentence independently. Thus, where in the sequence tagging model the preceding sentences influence the prediction for the current sentence, there is no interaction between sentences in the classification model. We use the latter model as a baseline in the evaluation stage.

5 EXPERIMENTS

To evaluate the quality of the automated summaries they are compared to the gold standard (author highlights) provided by the authors. The comparison is done on the top 4 ranked sentences, using `rouge-l-f`. The metric from here on will be referred to as `rouge-l-f@4`.

5.1 Settings

We initialise the model with `GLOVE` embeddings of size '100'. Unknown words receive an embedding which has been randomly initialised. Different models were trained with and without the ability to modify the embeddings (see results in Table 2).

Sentence Encoder. First the three proposed sentence encoders are tested. A number of the parameters are held constant due to results

[4]There are 334 unique All Science Journal Classification (ASJC) codes, each representing disciplines and sub-disciplines. More information on them can be found at https://service.elsevier.com/app/answers/detail/a_id/15181/supporthub/scopus/

from previous experimentation. The word embedding dimensionality is 100 across all the experiments and all LSTM cells within the sentence encoder have a hidden state of 100. For the CNN we use 25 filters of size $[1, 2, 3, 4]$. This means all three sentence encoders produce output vectors of size 100.

Sentence Extractor. The hidden state within the LSTM cell is consistently held at 128. The MLP layer is of size 50 before it is classified using a softmax layer. The ASJC embedding size is set to a size of 100 which is used in the additional features along with the title and abstract embeddings (which have a length of 100)

5.2 Training

Models are trained using weighted negative log-likelihood which has to be minimised.

$$L = -\sum_{s,y \in D} \sum_{i=1}^{n} \omega(y_i) \log p(y_i | y_{<i}, h) \tag{1}$$

The weights within the models are proportional to the number of positive and negative labels where $w(0) = 1$ and $w(1) = N_1/N_0$. Where N_n is the number of labels within the document for that classification.

The model was optimised using `ADAM` [12] and stochastic gradient descent, with a learning rate of 0.0001 and dropout across the model of 0.25. Additionally, gradient clipping was used to mitigate against the problem of vanishing gradient, this was set to 1. Models are trained for up-to 50 epochs, with early stopping if there has been no decrease in validation `loss` for 5 epochs.

6 RESULTS

The following section reports the results of the experiments outlined above. First the results for the varying sentence encoders are reported, with the best performing sentence encoder used to test the effectiveness of adding additional features to the model. The best performing model overall is submitted for human evaluation in sections 6.6.

All models are trained three times, and the results reported are created using a document level average across each of the runs. Additionally, an approximate randomisation test is used to report on statistic confidence in the results between models.[5]

6.1 Sentence Encoder

Results for the varying sentence encoders can be found in Table 2. The metrics reported are `rouge-l-f@4` against the author provided highlights in the validation set. Results indicate that the best performing models are those developed with the CNN based sentence encoder. This is consistent with the results when they are broken down by discipline. As one can see the results show limited variance between the different forms of sentence encoders. Moreover, within them all offer the same stability across disciplines. The same stable scores across the sentence encoders can be seen in [10], where no significant difference was reported between the models. This resulted in the MEAN embeddings being used. However, here the CNN based embedding with trainable word embeddings is used in the next experiment.

[5]https://github.com/Sleemanmunk/approximate-randomization

6.2 Additional Features

As proposed in sections 4.1.1 and 6.2 the model was modified with the addition of features to both the sentence encoder and sentence extractor. In Table 3 we show the results for the CNN based sentence encoder with trainable embeddings with the additional features. The inclusion of additional features with the sentence encoder results in a noticeable improvement in the `rouge-l-f@4` score, with computing articles seeing a maximum increase of 0.45. However, even though the modification of the sentence encoder improves the quality of the summaries, initialising the sentence extractor with the document level features degrades the quality of the summaries slightly. The document features on their own reduced the `rouge-l-f@4` by 0.05, though not statistically significant.

6.3 Baseline

The initial model discussed in section 4.2 with the addition of document and sentence level features significantly outperform the baseline model proposed in [5]. The baseline model classified each sentence separately from each other, meaning there is no addition of context in the classification from the surrounding sentences. The significant difference in the results from the two models indicates that the inclusion of more contextual information through the RNN is important. This is evidence that structure is an intrinsic feature within the model.

6.4 Structural Analysis

The training data-set is constructed using greedy sampling, with the majority of sentences coming from the 'Results' section of the articles. When looking at the distribution of the location of the predicted sentences within the document, one can see that there is indeed a dependency on the structure of the document on which sentences are selected. Within the top 4, 28.84% of sentences are from the 'Results' section, followed by sentences from the 'Introduction' which accounted for 19.79%. This distribution of sentences from across the article is not seen in the summarisation of news articles. [8] reported a strong lead-bias, where sentences from the beginning of the articles dominate the summaries.

To further investigate how much the model depends on the structure of the document, we train the model again with the sentences within each training document shuffled. The new model is then applied to the validation data-set which has not been shuffled. A significant drop in the metrics would then indicate that the structure of the document and the context which is learnt through the document level RNN is important.

Results can be seen in Table 4. As one can see there is a drop in performance across all disciplines, with the `rouge-l-f@4` reducing from from 22.19 to 20.75, a 9.35% reduction. The significant reduction across the board is similar to results reported in [10] for news, showing that the model is learning the position of the sentence within the document. This is understandable since scientific articles generally have a clear, intrinsic structure. Though when looking at the individual disciplines one can see that the size of the reduction varies across the board. Disciplines such as Economics and Computing, which are associated with a more varied writing style, have less of a reduction. This would indicate that for some

		Biology	Computing	Economics	All
CNN	False(*)	20.87	**23.01**	**21.92**	21.91
	True	**21.03**	22.97	21.61	**22.19**
MAN	False(*)	20.75	22.50	20.99	21.79
	True(*)	20.90	22.44	21.02	21.89
RNN	False(*)	20.85	22.60	21.31	21.86
	True(*)	20.49	22.33	20.97	21.51

Table 2: `rouge-l-f@4` scores for each of the six variations in sentence encoders. Models with a * indicate they are statistically worst than the best performing model ($p > 0.05$)

Sentence Features	Document Feature	Biology	Computing	Economics	All
False	False	21.03 (0.00)	22.97 (0.00)	21.61 (0.00)	22.19 (0.00)
	True	20.88 (-0.15)	23.34 (0.35)	21.49 (-0.12)	22.14 (-0.05)
True	**False**	**21.45** (0.42)	**23.42** (0.45)	**21.97** (0.36)	**22.37** (0.18)
	True(*)	20.57 (-0.46)	22.64 (-0.33)	21.05 (-0.56)	21.59 (-0.6)
Baseline					
Collins et al. [5]		13.12	13.42	16.61	12.96

Table 3: `rouge-l-f@4` scores for the CNN based sentence encoder model with additional features on both the sentence and document level. Models marked with a * when compared to the best model are statistically worse than the best performing model ($p > 0.05$). Value in brackets indicate difference between best performing model in Table 2

Shuffled	Biology	Computing	Economics	All
False	21.03	22.97	21.61	22.19
True	20.11	22.62	21.09	20.75

Table 4: `rouge-l-f@4` scores for the CNN based sentence encoder model without additional features.

disciplines the structure of documents is more important than for others.

6.5 Length Analysis

Author based highlights had on average 11.92 tokens, while we established that the sampled sentences (using `rouge-l-f`) are 15 tokens long. The best performing model extracted sentences with an average of 11.36 tokens. This is compared to the 32.43 tokens reported by Collins et al.. This shows that the model is not only looking at the text and location of the sentence but also the density of information within the sentence. Resulting in short highlights which are heuristically similar to the author provided highlights.

6.6 Human Evaluation

Computed metrics for text summarisation such as `rouge` only give an indication about the quality of the automated summary, thus human judgement was included. 7 human annotators were tasked with rating a set of 12 automated summaries, created using the best performing model (CNN sentence encoder, with additional sentence features), each document was assessed 3 times. The raters were not necessarily experts in the domains of the documents they assessed. The articles included in the test were a random sample from across academic disciplines.

Raters were asked to rate each set of summaries on a scale of 1 (low) to 4 (high) on four dimensions:

- **Simplicity**: are the sentences which have been selected simple to read or are they too long and using over-complicated language.
- **Informativeness**: do the sentences which have been selected inform the user about what is going on within the papers
- **Relevant**: are the sentences which have been selected relevant to the main findings of the paper
- **Diversity**: are all the sentence which have been selected covering the same points or is there diversity across the sentences.

In order to get a baseline, a number of author-generated summaries (gold standard) were included in the rating task. This allowed us to get a side by side comparison of how the best trained model performed compared to the gold standard author provided highlights. Inter-annotator agreement for the four dimensions was: 72.92, 66.67, 70.83, 54.17 respectively.

The raters assessed the author-provided summaries higher than the automated ones. However, the ratings for the automated highlights are not significantly worse than the author provided ones. The assessors judged that the automated summaries were both simple and diverse. It should be pointed out that there was limited inter-annotator agreement on the diversity, indicating potential misunderstanding of this dimension. The summaries were rated lower (but still favourably) on the informative and relevance scale.

7 DISCUSSION

Extracting simple automated summaries from content is challenging. The research presented in this paper demonstrates the ability to successfully apply neural extractive text summarisation methods

Author Highlight	Automated Summary
• The expression pattern of visfatin was ubiquitous in the various avian tissues. • Visfatin mRNA was most highly expressed in breast muscle and continuously decreased with increasing age in silky fowl. • Subcutaneous fat and visceral fat exhibited higher contents of visfatin mRNA in broiler chicken than those in silky fowl. • Visfatin fusion protein significantly increased the expression of adipocyte differentiation marker genes.	• Both subcutaneous fat and visceral fat exhibited higher contents of visfatin mRNA in broiler chickens than those in silky fowl. • 3T3-L1 adipocytes were treated with recombinant chicken visfatin and insulin. • Furthermore, visfatin fusion protein significantly increased the expression of adipocyte differentiation marker genes. • The visfatin mRNA levels continuously decreased with increasing age in silky fowl.

Table 5: Example author highlights and automated summary for the document "Characterization of the visfatin gene and its expression pattern and effect on 3T3-L1 adipocyte differentiation in chickens" (https://www.sciencedirec.com/science/pii/S0378111917306765)

Discipline	Dive	Info	Simp	Rele
Bio Science	2.81	2.49	2.93	2.65
Computing	2.95	1.57	2.95	1.86
Economic	3.00	2.67	3.67	2.33
All	2.90	2.37	2.83	2.51
Author highlights	3.01	2.78	2.81	3.2

Table 6: Results for human evaluation

to large scientific articles from a variety of domains. This work shows that moving from a binary classification based model to one which is sequential in nature can result in a significant uplift in the performance of the model. This means the model not only learns from the text but also takes the structure of the scientific article into account, bringing in local and global context to each prediction.

The positive results were observed across a variety of scientific domains (including social sciences). The robustness of the models across domains indicates that the model is potentially learning from common phrase structures seen across scientific disciplines. This would then explain why the inclusion of additional article and sentence-level features resulted in marginal increases in model performance. It would also explain the inclusion of sentences containing common phrases like 'Results can be found in section' in several summaries. The dependency on the structure of the articles seems to differ across disciplines, with a subject like Biology being more impacted in the sentence shuffle experiment. It should also be noted that sentences selected from the methods and results sections often include OOV words such as proteins and chemicals, thus potentially indicating that the sentence encoders were successfully picking up common patterns within the phrasing of the text.

Through the use of human evaluation, we showed that the automated highlights generated were comparable in quality to those submitted by the author. This could be interpreted two ways though, either the model is doing well, or the highlights submitted by the authors are not that good. Speaking to the raters and doing manual inspection of the summaries taught us that it could be a combination of the two. This can be seen in several examples where the author highlights in the validation set where taken directly from the article, with the model then predicting the correct sentences.

From a technical perspective one could argue that even though across the board the CNN achieves the best results, it might not the best model to deploy. A major drawback of this model is the length of time it takes to train compared to other models. Training the CNN model takes on average 24 hours, compared to 11 hours for the MEAN model.[6] The addition of sentence and article-level features resulted in marginal improvements, which may not have justified their inclusion as they again take time to compute.

8 CONCLUSION

To conclude, in this paper we presented an empirical analysis of using sequential models for extractive text summarisation. Results indicate that through the application of a basic RNN model, one can get summaries which are as good as those provided by the author of the paper. Depending on the disciplines of the paper there are varying degrees of reliance on the structure of the article. In further research we will look at the use of more complex sentence embeddings, which can take into account tokens such as protein and genes names which are currently treated as random embeddings.

REFERENCES

[1] 2014. *Evolution of reddit.* ACM Press, New York, New York, USA. https://doi.org/10.1145/2567948.2576943

[2] Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. SemEval 2017 Task 10 - ScienceIE - Extracting Keyphrases and Relations from Scientific Publications. *SemEval@ACL* (2017), 546–555. https://doi.org/10.18653/v1/S17-2091

[3] Jianpeng Cheng and Mirella Lapata. 2016. Neural Summarization by Extracting Sentences and Words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* Association for Computational Linguistics, Berlin, Germany, 484–494. https://doi.org/10.18653/v1/P16-1046

[4] Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. 2019. Structural Scaffolds for Citation Intent Classification in Scientific Publications. *arXiv.org* (April 2019). arXiv:1904.01608v2 [cs.CL] http://arxiv.org/abs/1904.01608v2

[5] Ed Collins, Isabelle Augenstein, and Sebastian Riedel. 2017. A Supervised Approach to Extractive Summarisation of Scientific Papers. *CoRR* (2017), 195–205. https://doi.org/10.18653/v1/K17-1021

[6] Günes Erkan and Dragomir R Radev. 2004. LexRank - Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research* 22 (2004), 457–479. https://www.scopus.com/inward/record.uri?partnerID=HzOxMe3b&scp=27344433526&origin=inward

[6] Trained on an AWS `ml.p3.16xlarge` instance

[7] Felix A Gers, Jürgen Schmidhuber, and Fred A Cummins. 2000. Learning to Forget - Continual Prediction with LSTM. *Neural Computation* 12, 10 (2000), 2451–2471. https://doi.org/10.1162/089976600300015015

[8] Kai Hong and Ani Nenkova. 2014. Improving the Estimation of Word Importance for News Multi-Document Summarization. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Gothenburg, Sweden, 712–721. https://doi.org/10.3115/v1/E14-1075

[9] Kokil Jaidka, Michihiro Yasunaga, Muthu Kumar Chandrasekaran, Dragomir Radev, and Min-Yen Kan. 2019. The cl-scisumm shared task 2018: Results and key insights. *arXiv preprint arXiv:1909.00764* (2019).

[10] Chris Kedzie, Kathleen R McKeown, and Hal Daumé III. 2018. Content Selection in Deep Learning Models of Summarization. *EMNLP* (2018). https://dblp.org/rec/conf/emnlp/KedzieMD18

[11] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. (Aug. 2014), 1746–1751. http://arxiv.org/abs/1408.5882v2

[12] Diederik P Kingma and Jimmy Ba. 2015. Adam - A Method for Stochastic Optimization. *ICLR* (2015). https://dblp.org/rec/journals/corr/KingmaB14

[13] Peder Larsen and Markus von Ins. 2010. The rate of growth in scientific publication and the decline in coverage provided by Science Citation Index. *Scientometrics* 84, 3 (2010), 575–603.

[14] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. *arXiv.org* (Aug. 2018). arXiv:1808.09602v1 [cs.CL] http://arxiv.org/abs/1808.09602v1

[15] Yi Luan, Mari Ostendorf, and Hannaneh Hajishirzi. 2017. Scientific Information Extraction with Semi-supervised Neural Tagging. *arXiv.org* (Aug. 2017). arXiv:1708.06075v1 [cs.CL] http://arxiv.org/abs/1708.06075v1

[16] Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Text. In *Proceedings of EMNLP 2004*. Association for Computational Linguistics, Barcelona, Spain, 404–411. https://www.aclweb.org/anthology/W04-3252

[17] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer - A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents. *AAAI* (2017), 3075–3081. https://www.scopus.com/inward/record.uri?partnerID=HzOxMe3b&scp=85030459977&origin=inward

[18] Shashi Narayan, Nikos Papasarantopoulos, Shay B Cohen, and Mirella Lapata. 2017. Neural Extractive Summarization with Side Information. *arXiv.org* (April 2017). arXiv:1704.04530v2 [cs.CL] http://arxiv.org/abs/1704.04530v2

[19] Jianmo Ni, Jiacheng Li, and Julian J Mcauley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. *EMNLP/IJCNLP* (2019), 188–197. https://doi.org/10.18653/v1/D19-1018

[20] Jianmo Ni, Zachary C Lipton, Sharad Vikram, and Julian J Mcauley. 2017. Estimating Reactions and Recommending Products with Generative Models of Reviews. *IJCNLP(1)* (2017). https://dblp.org/rec/conf/ijcnlp/NiLVM17

[21] Jianmo Ni and Julian McAuley. 2018. Personalized review generation by expanding phrases and attending on aspect-aware representations. In *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*. University of California, San Diego, San Diego, United States, 706–711. https://www.scopus.com/inward/record.uri?partnerID=HzOxMe3b&scp=85063144317&origin=inward

[22] Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*. Stanford University, Palo Alto, United States, Association for Computational Linguistics, Stroudsburg, PA, USA, 1073–1083. https://doi.org/10.18653/v1/P17-1099

[23] Benjamin W Stewart, Andy Rivas, and Luat T Vuong. 2017. Structure in scientific networks: towards predictions of research dynamism. *arXiv.org* (Aug. 2017). arXiv:1708.03850v1 [cs.SI] http://arxiv.org/abs/1708.03850v1

[24] Simone Teufel and Marc Moens. 2002. Articles Summarizing Scientific Articles: Experiments with Relevance and Rhetorical Status. *Computational Linguistics* 28, 4 (2002), 409–445. https://doi.org/10.1162/089120102762671936