Leveraging Information Bottleneck
for Scientific Document Summarization

Jiaxin Ju[1]**, Ming Liu**[*23]**, Huan Yee Koh**[1]**, Yuan Jin**[1]**, Lan Du**[1] **and Shirui Pan**[1]
[1]Faculty of Information Technology, Monash University, Australia
[2]School of Information Technology, Deakin University, Australia
[3]Zhongtukexin Co. Ltd. , Beijing, China
{jjuu0002,hykoh3}@student.monash.edu
m.liu@deakin.edu.au
{yuan.jin,lan.du,shirui.pan}@monash.edu

Abstract

This paper presents an unsupervised extractive approach to summarize scientific long documents based on the Information Bottleneck principle. Inspired by previous work which uses the Information Bottleneck principle for sentence compression, we extend it to document level summarization with two separate steps. In the first step, we use signal(s) as queries to retrieve the key content from the source document. Then, a pre-trained language model conducts further sentence search and edit to return the final extracted summaries. Importantly, our work can be flexibly extended to a multi-view framework by different signals. Automatic evaluation on three scientific document datasets verifies the effectiveness of the proposed framework. The further human evaluation suggests that the extracted summaries cover more content aspects than previous systems.

1 Introduction

Automatic text summarization is a challenging task of condensing the salient information from the source document into a shorter format. Two main categories are typically involved in the text summarization task, one is extractive approach (Cheng and Lapata, 2016; Nallapati et al., 2017; Xiao and Carenini, 2019; Cui et al., 2020) which directly extracts salient sentences from the input text as the summary, and the other is abstractive approach (Sutskever et al., 2014; See et al., 2017; Cohan et al., 2018; Sharma et al., 2019; Zhao et al., 2020) which imitates human behaviour to produce new sentences based on the extracted information from the source document. Traditional extractive summarization methods are mostly unsupervised, extracting sentences based on n-grams overlap (Nenkova and Vanderwende, 2005), relying on graph-based methods for sentence ranking (Mihalcea and Tarau, 2004; Erkan and Radev, 2004),

---
*Corresponding author

or identifying important sentences with a latent semantic analysis technique (Steinberger and Jezek, 2004). These unsupervised systems have been surpassed by neural-based models (Zaheer et al., 2020; Huang et al., 2021) in respect of performance and popularity, their encoder-decoder structures use either recurrent neural networks (Cheng and Lapata, 2016; Nallapati et al., 2016) or Transformer (Zhang et al., 2019; Khandelwal et al., 2019).

Chu and Liu (2019) developed an unsupervised auto-encoder model which attempts to encode and then reconstructs the documents with some properly designed reconstruction loss. However, as it tries to preserves every detail that helps to reconstruct the original documents, it is not applicable to long-document summarization settings. Recently, Ju et al. (2020) proposes an unsupervised non-neural approach for long document by building graphs to blend sentences from different text spans and leverage correlations among them. Nevertheless, none of the aforementioned works utilize explicit guidance to aid the model in summarizing a source text.

To this end, some works (Li et al., 2018; Liu et al., 2018; Zhu et al., 2020; Saito et al., 2020; Dou et al., 2021) explore the use of guided signals extracted from the input source document such as keywords, highlighted sentences and others to aid the model architecture in summarizing the input document. These works only utilize a single signal, and Dou et al. (2021) empirically showed that if multiple guided signals can be optimally exploited, the model could achieve even greater improvement to its summary outputs in the supervised neural summarization research space. Based on this finding, we propose a multi-view information bottleneck framework that can effectively incorporate multiple guided signals for the scientific document summarization task.

The original idea of information bottleneck (IB) principle (Tishby et al., 2000) in information the-
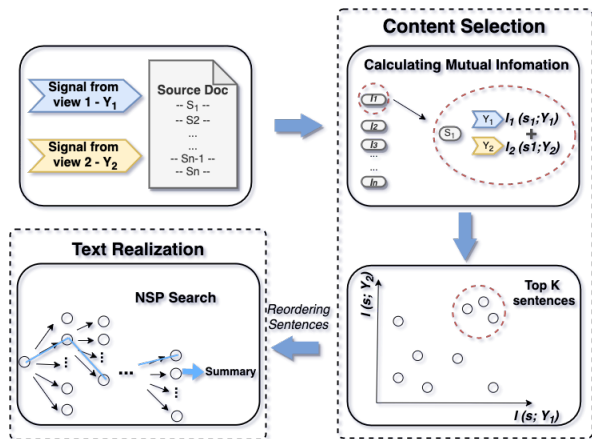
Figure 1: Our proposed multi-view information bottleneck framework. $I(s; Y)$ denotes the mutual information between sentence s and correlated signal Y, and NSP is short for Next Sentence Prediction task.

ory is to compress a signal under the guidance of another correlated signal. BottleSum (West et al., 2019) successfully applied IB to the summarization task for short document. Their model generates summary merely by removing words in each sentence while preserving all sentences without considering the importance of the sentences at a document level. It is not suitable for long scientific document summarization as it would preserve all sentences, resulting in significant redundancy.

In contrast, our framework applies IB principle on document level rather than sentence level, where pruning unrelated information will only work on the selected important sentences. In particular, at the content selection stage as shown in Figure 1, the signal that we seek to compress is the source document and the correlated signals are extracted from the source document using state-of-the-art language models. Followed by the text realization step where our proposed architecture conducts sentence search based on fluency to return the final extracted summaries.

This framework can be flexibly extended to multi-view architecture by incorporating more self-defined correlated signals. Our experiments on arXiv, PubMed (Cohan et al., 2018) and COVID-19 (Wang et al., 2020) show that our framework yields competitive performance compared to previous extractive summarizers. Despite the less satisfactory results for multi-view framework in our experiment, we believe it has fruitful potential for further study since the experiments from the work of Dou et al. (2021) have empirically shown that summarization

through multiple guided signals can achieve significant improvements to the system with a single signal.

2 Information Bottleneck for Unsupervised Summarization

2.1 Information Bottleneck principle

Information bottleneck (IB) principle (Tishby et al., 2000) naturally incorporates selection and pruning into compressing information. It compresses the input source S into $\tilde{S}$, which only preserves information related to signal Y, by minimizing the equation:

$$I(\tilde{S}; S) - \alpha I(\tilde{S}; Y) \quad (1)$$

where I denotes mutual information between two variables and the trade-off coefficient α controls the pruning term and relevance term. The term $I(\tilde{S}; S)$ is to prune irrelevant information, while $I(\tilde{S}; Y)$ enforces the model to retain more information correlated the label Y. For the summarization task, we define S to be the source document, $\tilde{S}$ is the output summary, and Y is the correlated signal.

2.2 Multi-view IB for scientific document summarization

To leverage the benefits of multiple guided signals, we seek to extend IB principles to effectively incorporate multiple guided signals. Recent work (Federici et al., 2020; Wu and Fischer, 2020) leverages the benefit of multi-view IB in other domains, so we extend this framework with multiple views by minimizing the following equation:

$$I(\tilde{S}; S) - \alpha \sum_{s \in \tilde{S}} I(s; Y_1) - \beta \sum_{s \in \tilde{S}} I(s; Y_2) \quad (2)$$

where Y_1 and Y_2 refer to two different views on the document content. In this equation, we consider the mutual information between sentences and the guided signal individually. In addition, the term $I(\tilde{S}; S)$ still is to remove redundant information, while $\sum_{s \in \tilde{S}} I(s; Y_1)$ and $\sum_{s \in \tilde{S}} I(s; Y_2)$ are to retain correlated information. The trade-off parameter α and β control the relationship between two views Y_1 Y_2 and the pruning term $I(\tilde{S}; S)$ respectively. However, there is no clear way to directly optimize a value for them without a supervised validation set so that we cannot directly compare the importance between two progresses. Instead, we

formalise this process as selection step and text editing step.

Followed by BottleSum(West et al., 2019), the equation (2) can be posed formally as learning a conditional distribution. As we extend their work to document level, the probability of the sentence selected by the system, $P(s)$, should be 1. Then the equation can be formulated as the following:

$$-log P(\tilde{S}) - \sum_{s \in \tilde{S}} \Big[\alpha P(Y_1|s) \log P(Y_1|s) \\ + \beta P(Y_2|s) \log P(Y_2|s) \Big] \quad (3)$$

Thus, the content selection is to keep relevant information by maximizing $P(Y_1|s)$ and $P(Y_2|s)$ while the text editing step is to prune irrelevant sentences by optimizing $P(\tilde{S})$. In our framework, we define Y_1 to be the document categories (e.g. cs, math), and Y_2 to be a keyphrase list of the specific article. The equation eventually can be rewritten as[1]:

$$-log P(\tilde{S}) - \sum_{s \in \tilde{S}} \Big[\alpha P(Y_1|s) \log P(Y_1|s) \\ + \beta \sum_{y \in Y_2} P(y|s) \log P(y|s) \Big] \quad (4)$$

where y is the keywords in the extracted keywords list Y_2. Hence, our goal is to maximize $P(Y_1|s)$ and $P(y|s)$ while optimizing $P(\tilde{S})$.

2.3 Proposed algorithm

To illustrate how our frameworks are learned based on the IB principle, we divide Equation (4) into two parts and develop an algorithm for each part. The content selection algorithm corresponds to the second term. The algorithm below shows a generalized framework that can be extended to include more than two signals, $Y = \{Y_1, Y_2, ..., Y_n\}$. The implementation details of $Y_i(s)$ will be explained in the section 4.2. The higher the score a sentence gains, the stronger the correlation with the guided signal(s) and the higher the probability it will be included in the output summary.

For text realization algorithm, the candidate sentence set selected from content selection step is firstly reordered in terms of the sentence original position in the source document. Then we use SciBERT (Beltagy et al., 2019) to apply the next

---

[1]Detailed derivations of this formula can be found in the Appendix, and our code: https://github.com/Jiaxin-Ju/Unsupervised_IB_Summ

---

Algorithm: Content selection & Text Realization

Require: Document D, signal set $Y = \{Y_1, Y_2, ..., Y_n\}$, position information Pos, and a language model LM
Content Selection:
1: $S_D \leftarrow$ split doc D * full sentence set
2: **for** each s in S_D **do**
3:    **if** len(s) in length constraint **then**
4:      **for** each Y_i in Y **do**
5:        $P(Y_i|s) \leftarrow Y_i(s)$
6:        $Score(Y_i) = P(Y_i|s) \times log(P(Y_i|s))$
7:      $Score(s) = \sum_{i=1}^{n} Score(Y_i)$
8: **sort** S_D based on $Score(s)$ **then**
9:    $S_t \leftarrow$ top N sentences in S_D
Text Realization:
1:    **sort** S_t based on Pos **then**
2:    **for** m in $(0 ... len(S_t) - 1)$ **do**
3:      **for** n in $m + 1 ... len(S_t)$ **do**
4:        $P_{m,n} \leftarrow LM(S_t[m], S_t[n])$
5:    matrix $M \leftarrow P_{m,n}$
6:    **for** s in $\{first\ k\ sents\}$ of sorted S_C
         set s as the start sentence **do**
7:      best sent path $\leftarrow$ **Search**(s, M)
8:    best summary path $\leftarrow$ k best sent paths
9:    $S_{sum} \leftarrow$ best summary path
10:    **return** S_{sum}

---

sentence prediction (NSP) task, each sentence is evaluated against the sentence appeared before it to determine the likelihood that these two sentences are consecutive. Similar idea based on BERT NSP task has been proposed by (Bommasani and Cardie, 2020) to measure a summary's semantic coherence and fluency. Taking fluency of the summary into account, this searching algorithm aims to find the most likely sentence combination as the candidate summary, and the best sentence combination will be selected from these k candidate combinations. Here we implemented greedy search and beam search respectively for model performance comparison. The greedy search algorithm is started from the first sentence, then we find the sentence combination with the highest next sentence probability in each window. For beam search, since the best sentence combination may not start from the first sentence, we perform it for the first k sentences of the candidate sentence set.

3 Experiment

3.1 Datasets

Additional to the widely-used arXiv and PubMed datasets (Cohan et al., 2018), we also make use of the COVID-19 scientific paper dataset (Wang et al., 2020). The dataset statistics can be seen in Table 2.

3.2 Experiment Setup

Content selection We define a list of keyphrase extracted by RAKE (Rose et al., 2010) as the cor-

	arXiv			PubMed			COVID-19		
	R-1	**R-2**	**R-L**	**R-1**	**R-2**	**R-L**	**R-1**	**R-2**	**R-L**
Oracle	42.37	19.23	39.26	48.10	26.71	45.04	44.61	23.02	40.16
Lead-3	25.68	5.92	20.53	26.71	8.78	23.61	20.88	6.29	18.66
Unsupervised Models (rerun)									
LSA(Steinberger and Jezek, 2004)	32.55	7.54	25.72	34.60	10.07	28.64	30.18	7.12	23.28
SumBasic(Vanderwende et al., 2007)	30.48	6.70	26.34	**37.76**	11.68	30.19	30.46	7.61	24.63
TextRank (Barrios et al., 2016)	31.34	8.68	23.96	33.12	11.84	28.12	27.76	8.60	22.51
SciSummPip (Ju et al., 2020)	30.97	_10.13_	25.61	36.30	11.96	28.98	29.87	8.27	23.97
Our Work									
keywords only	_33.40_	**10.33**	**27.66**	_36.82_	12.18	29.99	31.61	8.75	24.10
Keywords + beamSearch	**33.64**	9.96	_26.67_	36.55	**13.36**	_30.83_	**33.04**	9.96	_25.22_
keywords + greedySearch	32.13	9.57	26.00	36.19	_13.19_	30.79	_33.00_	_9.87_	**25.26**
MultiViewIB + beamSearch	32.66	9.05	25.54	36.20	12.98	30.36	32.11	9.32	24.31

Table 1: Comparisons with unsupervised extractive models on three scientific datasets. The best F1 results are in **boldface**, and the second highest scores are in _italic_. The implemented TextRank (Barrios et al., 2016) improves the original performance (Mihalcea and Tarau, 2004) by modifying the similarity function to Okapi BM25.

		Median doc length		**Median abstract length**	
Datasets	**test docs**	words	sents	words	sents
arXiv	6,440	4319	203	142	6
PubMed	6,658	2293	82	190	7
Covid19	5,178	3906	140	231	8

Table 2: Elementary data statistics for the test sets of three datasets. We select approximately number of papers as the COVID-19 test set.

related signal for single view, while the multi-view framework incorporates the document category as another view. Top 10 keyphrases are extracted, and sentences and keyphrases are then mapped into high dimensional space by averaging the output from SciBERT (Beltagy et al., 2019). We assume the sentence with higher similarity to the keyphrases are more likely to associate with the defined signal, and the score is the summation of the cosine similarity between the sentence and each keyphrase. For multi-view framework, we use Longformer (Beltagy et al., 2020) that is pre-trained with 100 classes on the kaggle arXiv dataset[2] to obtain the $P(Y|s)$ for each sentence. In the pre-training process, we utilize the large model of Longformer and we set the learning rate as 1e-5, batch size as 4, epoch as 4, hidden dropout as 0.05 and the hidden size as 1024. 50 sentences with higher score will be selected for the next step.

Text Realization   For NSP task, we continue to use SciBERT (Beltagy et al., 2019) to obtain the likelihood of two adjacent sentences. We implement greedy search and beam search respectively for model performance comparison. The greedy

[2]https://www.kaggle.com/Cornell-University/arxiv

search algorithm is started from the first sentence (k=1), then we find the sentence combination with the highest next sentence probability in each window. We set the window size to 3 then slide the window by one sentence. For beam search, since the best sentence combination may not start from the first sentence, we perform it for the first 5 (k=5) sentences of the candidate sentence set and we set the beam size to 5. The number of sentences in the generated summary is 10.

3.3   Experiment Results

Results on scientific datasets   We compare our framework with unsupervised summarization models as shown in Table 1. We rerun these models and the number of sentences in the generated summary from all models is 10. Our models achieve the highest R-1 on arXiv and the highest R-2 on PubMed. On the COVID-19, the keywords+beam search setting achieves the highest score. SciSummPip (Ju et al., 2020) is a hybrid method that compresses and rewrites extracted sentences by building a word-relational graph, so it is likely to have more bigrams that match the reference summary. SumBasic (Vanderwende et al., 2007) tends to extract the sentence that contains more high frequency word so that it achieves a higher R-1 on PubMed. The comparison among our frameworks shows that single view settings performs better than multi-view setting, and the beam search algorithm is better than the greedy search algorithm. While we achieve better scores than baseline results, the performance differences are not significant. Thus, to investigate the effectiveness of our proposed framework, we further conduct the position analysis and human
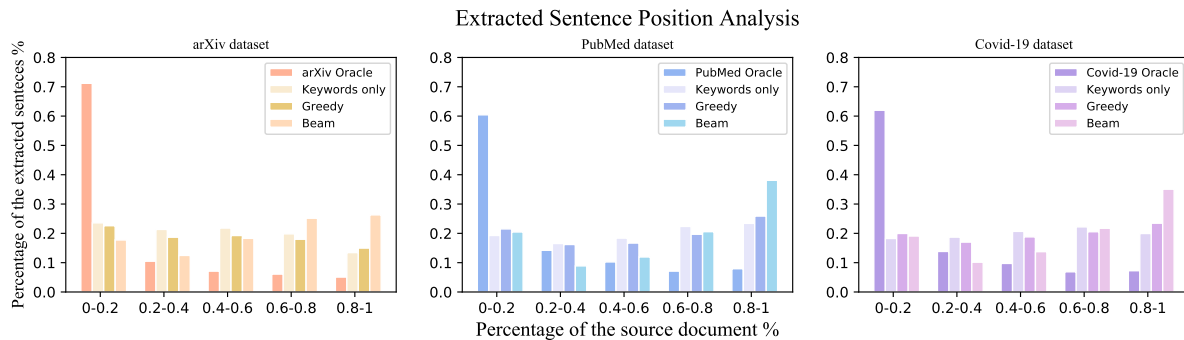
Extracted Sentence Position Analysis

Figure 2: The sentence position distribution of the extracted summaries and the oracle summaries.

Model	Flu	Fai	Cov	Con
Oracle	4.2	4.6	4.0	2.4
Keywords	2.8	4.0	4.0	3.0
Keywords + beamSearch	3.4	4.0	4.2	3.4
Keywords + greedySearch	3.4	4.2	3.8	3.2

Table 3: Human Evaluation. Flu, Fai, Cov and Con refer to Fluency, Faithfulness, Coverage and Conciseness.

evaluation on single view settings.

Sentence position analysis   Our position analysis is shown in Figure 2., the oracle summaries are mostly extracted from the beginning of the source document, while summaries extracted by our models are from all the sections within the source document. Achieving a higher ROUGE score can prove that the model captures unigram/bigram appeared in the reference summary, but the more important thing is the extracted summary can concisely cover most/all of the key information in each section of the original article for the reader and our model seems to achieve this significantly better than the oracle summary. To prove this hypothesis, we conduct a thorough human analysis.

Human analysis   We randomly sample 50 documents from the COVID-19 dataset and conduct human evaluation against four criteria: fluency, faithfulness, coverage and conciseness. For each article we compare summaries generated from 4 frameworks with the true summary, and the human annotators are asked to blind rate these summaries on a 1-5 point scale (1 is the worst and 5 is the best). The average performance of each model is shown in Table 3. Even though keywords+beamSearch setting does not significantly perform better than others in terms of ROUGE score, it receive higher human ratings. In addition, oracle summary perform better

Figure 3: An Example taken from COVID-19 dataset. Text in the same color indicates the contents they described are the same.

on fluency and faithfulness but it contains more unnecessary sentences. Figure 3 shows an example[3] of the abstract and the system summaries.

4   Conclusion and Future work

In this paper, we proposed an unsupervised framework based on IB principle for long document summarization. Our framework employs a two-steps system where content selection is guided by defined signal(s) and is followed by a text realization step where a pre-trained language model conducts sentence search to return final summaries. Experiments on three scientific show the effectiveness of our framework. Further human analysis suggests that the extracted summaries exhibit more coverage. Despite the less satisfactory results for multi-view framework in our experiment, we believe it has fruitful potential for further study.

---

[3]Original paper can be found at https://arxiv.org/pdf/2007.13933.pdf

References

Federico Barrios, Federico López, Luis Argerich, and Rosa Wachenchauzer. 2016. Variations of the similarity function of textrank for automated summarization. *ArXiv preprint*, abs/1602.03606.

Iz Beltagy, Arman Cohan, and Kyle Lo. 2019. Scibert: Pretrained contextualized embeddings for scientific text. *ArXiv preprint*, abs/1903.10676.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *ArXiv preprint*, abs/2004.05150.

Rishi Bommasani and Claire Cardie. 2020. Intrinsic evaluation of summarization datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8075–8096, Online. Association for Computational Linguistics.

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)*, pages 484–494, Berlin, Germany. Association for Computational Linguistics.

Eric Chu and Peter J. Liu. 2019. Meansum: A neural model for unsupervised multi-document abstractive summarization. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 1223–1232. PMLR.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Peng Cui, Le Hu, and Yuanchao Liu. 2020. Enhancing extractive text summarization with topic-aware graph neural networks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5360–5371, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. GSum: A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842, Online. Association for Computational Linguistics.

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.

Marco Federici, Anjan Dutta, Patrick Forré, Nate Kushman, and Zeynep Akata. 2020. Learning robust representations via multi-view information bottleneck. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online. Association for Computational Linguistics.

Jiaxin Ju, Ming Liu, Longxiang Gao, and Shirui Pan. 2020. Monash-summ@LongSumm 20 SciSummPip: An unsupervised scientific paper summarization pipeline. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 318–327, Online. Association for Computational Linguistics.

Urvashi Khandelwal, Kevin Clark, Dan Jurafsky, and Lukasz Kaiser. 2019. Sample efficient text summarization using a single pre-trained transformer. *ArXiv preprint*, abs/1905.08836.

Chenliang Li, Weiran Xu, Si Li, and Sheng Gao. 2018. Guiding generation for abstractive text summarization based on key information guide network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 55–60, New Orleans, Louisiana. Association for Computational Linguistics.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3075–3081. AAAI Press.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Ani Nenkova and Lucy Vanderwende. 2005. The impact of frequency on summarization. *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005*, 101.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, 1:1–20.

Itsumi Saito, Kyosuke Nishida, Kosuke Nishida, and Junji Tomita. 2020. Abstractive summarization with combination of pre-trained sequence-to-sequence and saliency models. *ArXiv preprint*, abs/2003.13028.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Eva Sharma, Luyang Huang, Zhe Hu, and Lu Wang. 2019. An entity-driven framework for abstractive summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3280–3291, Hong Kong, China. Association for Computational Linguistics.

Josef Steinberger and Karel Jezek. 2004. Using latent semantic analysis in text summarization and summary evaluation. *Proc. ISIM*, 4:93–100.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. *arXiv preprint physics/0004057*.

Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. 2007. Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing & Management*, 43(6):1606–1618.

Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Doug Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Michael Kinney, Yunyao Li, Ziyang Liu, William Merrill, Paul Mooney, Dewey A. Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex D. Wade, Kuansan Wang, Nancy Xin Ru Wang, Christopher Wilhelm, Boya Xie, Douglas M. Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. 2020. CORD-19: The COVID-19 open research dataset. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, Online. Association for Computational Linguistics.

Peter West, Ari Holtzman, Jan Buys, and Yejin Choi. 2019. BottleSum: Unsupervised and self-supervised sentence summarization using the information bottleneck principle. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3752–3761, Hong Kong, China. Association for Computational Linguistics.

Tailin Wu and Ian S. Fischer. 2020. Phase transitions for the information bottleneck in representation learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Wen Xiao and Giuseppe Carenini. 2019. Extractive summarization of long documents by combining global and local context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3011–3021, Hong Kong, China. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *ArXiv preprint*, abs/2007.14062.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069, Florence, Italy. Association for Computational Linguistics.

Jinming Zhao, Ming Liu, Longxiang Gao, Yuan Jin, Lan Du, He Zhao, He Zhang, and Gholamreza Haffari. 2020. Summpip: Unsupervised multi-document summarization with sentence graph compression. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 1949–1952. ACM.

Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. 2020. Boosting factual correctness of abstractive summarization with knowledge graph. *arXiv e-prints*, pages arXiv–2003.

A Derivation of formula

$$I(\tilde{S}; S) - \alpha I(\tilde{S}; Y_1) - \beta I(\tilde{S}; Y_2)$$

$$= I(\tilde{S}; S) - \alpha \sum_{s \in \tilde{S}} I(s; Y_1) - \beta \sum_{s \in \tilde{S}} I(s; Y_2)$$

Followed the previous work (West et al., 2019), we can rewrite the equation into an alternate form as shown below:

$$p(S, \tilde{S})pmi(\tilde{S}; S) - \alpha \sum_{s \in \tilde{S}} P(Y_1, s)pmi(s; Y_1) - \beta \sum_{s \in \tilde{S}} p(Y_2, s)pmi(s; Y_2)$$

where $pmi(x, y) = \frac{p(x,y)}{p(x)p(y)}$ denotes pointwise mutual information.

$$= P(S, \tilde{S}) \log \frac{P(S, \tilde{S})}{P(S)P(\tilde{S})} - \alpha \sum_{s \in s} P(Y_1, s) \log \frac{P(Y_1, s)}{P(Y_1)P(s)} - \beta \sum_{s \in \tilde{S}} P(Y_2, s) \log \frac{P(Y_2, s)}{P(Y_2)P(s)}$$

$$= P(\tilde{S}|S)P(S) \log \frac{P(\tilde{S}|S)}{P(\tilde{S})} - \alpha \sum_{s \in \tilde{S}} P(Y_1|s)P(s) \log \frac{P(Y_1|s)}{P(Y_1)} - \beta \sum_{s \in \tilde{S}} P(Y_2|s)P(s) \log \frac{P(Y_2|s)}{P(Y_2)}$$

$P(\tilde{S}|S)$=1 for chosen summary, $P(\tilde{S})$, $P(s)$, $P(Y_1)$ and $P(Y_2)$ are constant.

$$= P(\tilde{S}|S)P(S) \log \frac{P(\tilde{S}|S)}{P(\tilde{S})} - \sum_{s \in \tilde{S}} P(s) \left[\alpha P(Y_1|s) \log \frac{P(Y_1|s)}{P(Y_1)} + \beta P(Y_2|s) \log \frac{P(Y_2|s)}{P(Y_2)} \right]$$

$$= C_1 log \frac{1}{P(\tilde{S})} - \sum_{s \in \tilde{S}} C_2 \left[\alpha P(Y_1|s) \log P(Y_1|s) + \beta P(Y_2|s) \log P(Y_2|s) - C_3 \right]$$

$-log(p_G)$ is constant, so $P(Y_1|s) \times \log P(Y_1)$ is constant. For each sentence, $P(Y_1|s)$ will be scaled up or down in the same proportion, because $P(Y_1)$ and $P(Y_2)$ are constant and Log is a monotonically increasing function.

$$= -log P(\tilde{S}) - \sum_{s \in \tilde{S}} \left[\alpha P(Y_1|s) \log P(Y_1|s) + \beta P(Y_2|s) \log P(Y_2|s) \right]$$