Unsupervised Summarization Approaches for Slide Generation

1st Maria João Simões Costa
Instituto Pedro Nunes, LIS
Portugal
mariacosta@student.dei.uc.pt

Hugo Amaro
Instituto Pedro Nunes, LIS
Portugal
hamaro@ipn.pt

Hugo Gonçalo Oliveira
CISUC, DEI
CISUC, DEI, University of Coimbra
Portugal
hroliv@dei.uc.pt

***Abstract*—Presentation slides are essential for introducing new topics in an educational context, but they take time to create. Automatic slides generation aims at minimising the time required for preparing slide decks. This is often accomplished in two main steps: summarizing a document and organizing the result into slides. This paper focuses on the former, particularly in unsupervised approaches, in order to determine whether these approaches, which do not require training data, are viable when compared with supervised ones. For that, six methods are assessed, based on the ROUGE metric, when applied to two datasets of scientific papers and the slides used for presenting them. Despite being unsupervised, the performance of the tested methods is in line with the state-of-the-art, suggesting that they can be regarded as a good alternative, especially when training data is not available.**

***Index Terms*—Summarization, Automatic Slide Generation, ROUGE, LSA, TF-IDF, QueSTS, LexRank, TextRank, Lexical Chains**

I. Introduction

Technology is becoming increasingly important in today's world, with computational applications in practically every aspect of people's lives. This is the case of education, where slide shows are one of the most widely used tools, when it comes to introducing new topics to an audience. Creating slides, on the other hand, can be a complex and time-consuming task, because it is often necessary to read and summarise several documents related on the target topic. To help with the process, machine learning and natural language processing can be explored, allowing teachers, and presenters in general, to make better use of their time by only having to edit certain elements rather than preparing full presentations from scratch.

Automatic slide generation has generally two main steps [11], [16], [18]: summarization, where a shorter version of a source document is produced, while keeping the most important topics; and organizing the said text into slides. This paper focuses mainly on the first step, but it also presents a simple way to approach the second step. Six popular summarization methods are compared in this work: LexRank [3], TextRank [10], TF-IDF [6], [9], QueSTS [15], LSA [2], and Lexical Chains [14]. All of them are extractive methods, meaning that they select the most relevant sentences from a text without adding new words or changing the phrase structure. This is in opposition to abstractive approaches [5], where source text can be paraphrased, while transmitting the same meaning.

The explored methods have been extensively used in automatic summarization tasks, but not so much for slide generation. In order to test them in this specific use case, we resort on two available datasets of scientific papers and their slides, produced by humans, namely PS5K [13] and SciDuet [18]. The text from the papers is given as input to the approaches and the resulting summary is compared with the text of the human-produced slides. This is done with ROUGE [8], a popular set of metrics commonly used for evaluating automatically-created against ideal summaries, based on their overlap.

With this, we aim to better understand to what extent these methods are a viable alternative to supervised approaches. Despite the availability of the aforementioned datasets, they are too specific for their domain. In most cases, there are not enough examples of documents and their slides to be used as training data, i.e., not only on the target domain, but also not in the target language. Relying on unsupervised methods would enable summarization for presentation slides out-of-the-box, to any document, on any domain, style, as well as many languages.

Besides comparing the six methods, we look at performances reported in the literature for supervised approaches, to find that the unsupervised methods perform closely, and some even outperform them in a few of the tested ROUGE metrics. This was the case with TextRank in one of the ROUGE metrics for the PS5K dataset, whereas every method was superior for SciDuet in also one metric.

This document is structured in six sections: Related Work, which overviews work using the PS5K and SciDuet datasets; Experimentation, which describes the datasets, methods and evaluation metric used; Results, which reports and discusses the main findings in the summarization step; Slide Generation, which explains how slides are created having as a starting point a summary; Conclusion, which stresses the main conclusions and discusses future work.

II. Related Work

In this work two different datasets were used in order to test the unsupervised summarization methods: PS5K [13] and SciDuet [18], released as part of two recent studies on summarization for slide generation.

One of such studies [11] presented an extractive summarization method with three main steps: sentence labelling, sentence scoring, and sentence ranking. An important feature is that the document is first split in non-overlapping text windows. Within them, sentences are labelled having in mind the maximisation of the ROUGE score [8], when compared to the gold summaries in PS5K. After this, embeddings are learned for each sentence and document, with a bi-LSTM encoder (simple and with attention). Finally, sentences are ranked by a function that considers their position in the document, their salience, their novelty (when compared to the remaining sentences), and the similarity of their content to the gold summaries. For the best configuration, ROUGE-1, 2 and L scores of 0.48, 0.12, and 0.24, respectively, are reported.

Another interesting study [18], for which SciDuet was created, tackles automatic slide generation as a question-answering problem. Given the title for each slide, a suitable question is made to the document, in order to retrieve suitable content. This is done across three modules, respectively for: building a tree of titles; computing the similarity of text snippets, represented by dense vectors; asking questions, through a BART [7] model. The selection of figures and tables, through the similarity of their caption and the text, may follow the previous. For the best configuration, ROUGE-1, 2 and L scores of 0.20, 0.05, and 0.19, respectively, are reported.

Both of the previous approaches are supervised, and therefore need data. Due to the scarcity of datasets, their authors had to create their own datasets. Even if these datasets can be helpful for training and testing approaches for the automatic generation of slides, they are not ideal for domains and styles other than scientific papers, a very specific style and not an easy one, since these papers frequently include complex terms, formulas, and nuances. This is also why we test unsupervised approaches, namely LexRank [3], TextRank [10], TF-IDF [6], [9], QueSTS [15], LSA [2], and Lexical Chains [14], further described in section III. Despite being quite popular for automatic summarization, when it comes to slide generation, have not been sufficiently explored.

Moreover, as it happens to most related work, the described works rely exclusive on the ROUGE metrics to evaluate their summaries.

III. Experimentation

This section briefly describes the two datasets used, the six unsupervised methods for summarization that were assessed, and the metric that was used to evaluate the summaries.

A. Evaluation Data

All the explored methods aim to summarise texts. However, as our focus is on summarisation for presentation slides, we test the selected methods on two datasets recently made available, SciDuet [18] and PS5K [13], which include published scientific papers paired with human-produced slides used for presenting them. The datasets presents papers in the field of computer and information science, Machine Learning, Neural Information Processing Systems, and Computational Linguistics. Still, we should stress that we are not assessing the actual quality of the slides, only of the text that might be included in them. Summaries for slides have some characteristics that we are not considering specifically. For instance, they often include single statements or lists of items, lacking context, to be filled by the speaker, or by information in images or diagrams. Moreover, the tested methods are all for extractive summarization, while it is normal that text in the papers is changed to better fit the presentation.

Both datasets are divided into train, validation and test. Since we are using only unsupervised methods, only the test portion is used for evaluation. This encompasses 81 and 250 papers and their slides, respectively for SciDuet and for PS5K.

B. Methods

The six tested methods were: LexRank [3], TextRank [10], TF-IDF [6], [9], QueSTS [15], LSA [2], and Lexical Chains [14]. QueSTS and TF-IDF were chosen due to their previous utilization on the state-of-the-art for automatic slide generation. The others were chosen based on their popularity in the context of text summarization. Below is a brief explanation of each.

TF-IDF [1] (term frequency-inverse document frequency) [6] is a statistical method that computes the importance of a term in a document, considering its frequency in the document (TF) and in a collection of documents (DF). The former contributes positively, but the more documents a term occurs in, the lower its importance in any document is. When applied to summarization, TF-IDF is computed for every word, and the sentences with the greatest accumulative score are the regarded as the most important, i.e., to be included in the summary.

LSA [2][2] analyses relationships between a set of documents and the terms they contain. For our experiments, we have adopted a method [17] that selects the longest sentences for the summary. The length of a sentence is determined by considering concepts with indices lower than or equal to the provided dimension, which is seen in a concept-sentence matrix. The value is then multiplied by the second matrix produced by Singular Value Decomposition of a term-document matrix to emphasise the most significant concepts.

TextRank [10][2] builds a graph with sentences as nodes and the cosine similarity between sentences as edges. The resulting summary includes the higher-ranked sentences, weighted according to the weights.

LexRank [3][3] is similar to TextRank, except for the calculation of similarity, where a modified cosine with IDF (measure that seeks to decrease the weight of words that are frequent in many documents) is used and for the selection of the important sentences.For its implementation, a set of documents is required for computing the IDF. We used the training portions of the datasets, but any corpus can be used.

Another graph-based method [16] summarises academic papers with a query-specific summarizer—QueSTS [15]. Its implementation is inspired by [1]. It creates a graph similar to the previous implementations. Then, from each node, a

[1]Implementation inspired by: https://towardsdatascience.com/natural-language-processing-feature-engineering-using-tf-idf-e8b9d00e7e76

[2]https://pypi.org/project/sumy/

[3]https://github.com/crabcamp/lexrank

contextual tree is built for each query term. The node is added to the tree alongside its neighbours until it reaches the node with the query term. In the end, the trees of all the query terms for a node are merged into a $SGraphr$. All the $SGraphs$ are ranked by a scored model and the one with the highest rank is the summary.

Lexical Chains (implementation inspired by [14]) are sequences of semantic related words. It uses WordNet [4], which is structured on synsets, which are groups of words with the same meaning. This method involves creating chains, each of which contains a group of words (only nouns were considered) with the same meaning. For each chain that respects the following equation (Strong Chains), the first sentence containing this chain is added to the summary: $Score(Chain) > AVG(Scores) + ratio \times STD(Scores)$. The score of a chain is calculated using the word frequencies and the number of words in a chain. All the sentences with a score greater than the average of sentences scores are also added to the summary. The score is given by the nouns in strong chains and the sentence length.

C. Metrics

Our evaluation relied on the ROUGE [8] metric, widely used across summarization and other text generation problems. ROUGE compares the automatic and the reference summary according to different metrics, each one with a corresponding recall, precision and F-score. Among them, we use: ROUGE-N, with N either 1 or 2, corresponding to the overlap of sequences of N words; ROUGE-L, which measures the longest sequence of words (LCS) shared by both summaries; and ROUGE-W, similar to ROUGE-L, but also tracking the lengths of consecutive matches, in addition to the length of the LCS.

IV. RESULTS

This section presents and discusses the performance of the tested methods in the two datasets, according to the adopted ROUGE metrics. For each dataset and method, tests were conducted with different ratios – i.e., the lower the ratio, the shorter the summary – as well as applying preprocessing or not.

Every metric was computed for the same ratio values, except for TF-IDF, where sentences with a higher score than the average are put in the summary, regardless of their size; Lexical Chains (LC), where the ratio is a multiplication value used to determine the value that a chain must have to be considered strong; and for QueSTS, where the ratio corresponds to the number of leaves a tree can have.

A. Results in PS5K

Table I shows the ROUGE-1, 2, L and W scores in the PS5K dataset. For all methods, several tests were conducted with different ratios. Yet, due to space limitations, the table includes only the best and worst scores. Coincidentally, and with the exception of QueSTS and Lexical Chains (LC), these correspond to the lowest (LR) and highest ratios (HR) tested. For QueSTS, LR is 3 leafs and HR is 11. For Lexical Chains, LR is 0.5 and HR is 1. For the other methods, LR is 0.1 and HR is 0.4.

For example, looking at the scores of LexRank, we see that LR has a greater precision than HR, which allowed for a higher score. However, recall is lower. This shows that even though a higher ratio, i.e, a bigger summary results in a greater fraction of relevant sentences retrieved, there are also more that are not relevant when compared to the quantity in the smaller summary. So, despite having less relevant sentences, the shortest summaries end up having higher relevance as a whole. This is true for every metric, and for every method, except LC and QueSTS.

Even though in QueSTS a lower ratio has greater precision and lower recall, the latter value is much lower than for other methods. For example, for ROUGE-1, the recall is only 0.085 for QueSTS, with the next worst belonging to LC HR with a value of 0.190, thus resulting in a better F-score for the higher ratio experiences. This shows that the LR summary is too small and so, even if the sentences that are retrieved are relevant, they are very few and more are required. This happens because this method handles the ratio differently: a low ratio here corresponds to a shorter summary than for the other methods. So, while the summary in the other methods has a good size, in QueSTS it is too short, and so needs a higher ratio.

Something similar happens in LC, which, contrarily to other methods, with a LR has a higher recall and a lower precision, resulting in better summaries with a HR. Such as QueSTS, due to the different ratio utilization, this method's lower ratio is too low and so it also requires a higher ratio.

In a more general view, we can see that the best method for ROUGE-1 is LSA LR (followed closely by TextRank LR). For ROUGE-2, the best is TextRank LR and LC HR, for ROUGE-L and W the best is also TextRank LR. So, in this dataset the best performing method is TextRank.

ROUGE-1 and 2 scores achieved are all below the best supervised method reported for PS5K in related work [13], respectively 0.48 and 0.12. For ROUGE-L, however, the best reported score is 0.238, which is in line with LSA LR (0.235) and LexRank LR (0.233) and is outperformed by TextRank LR (0.247). Even though ROUGE-L is a limited metric, this suggests that unsupervised methods should be regarded as interesting alternatives to explore in slide generation.

B. Results in SciDuet

In SciDuet, much like PS5K, tests with lower ratios have, mostly, reached better ROUGE scores (see Table II), with the precision better and the recall worse than the experiments with higher ratio. The only difference lies in ROUGE-2, which has better F-scores with higher ratios, for every method. This suggests that SciDuet is better suited for shorter summaries, with not as many important sentences as the summaries with higher ratios, but showing more relevance, i.e., less sentences that do not matter.

Furthermore, with this dataset, for ROUGE-1, the best method is TF-IDF, while for ROUGE-2 and L is QueSTS HR.

Method	ROUGE-1			ROUGE-2			ROUGE-L			ROUGE-W		
	R	P	F	R	P	F	R	P	F	R	P	F
TF-IDF	0.346	0.175	0.233	0.080	0.039	0.052	0.314	0.160	0.212	0.089	0.066	0.076
TextRank HR	**0.444**	0.103	0.167	**0.143**	0.032	0.052	**0.412**	0.096	0.155	**0.118**	0.039	0.059
TextRank LR	0.323	0.209	0.255	0.080	0.050	**0.062**	0.316	0.202	**0.247**	0.091	0.086	**0.088**
QueSTS HR	0.206	0.306	0.246	0.040	0.061	0.048	0.189	0.284	0.227	0.056	0.125	0.077
QueSTS LR	0.085	**0.403**	0.140	0.013	0.071	0.022	0.079	**0.379**	0.131	0.026	**0.189**	0.046
LexRank HR	0.438	0.104	0.168	0.139	0.032	0.052	0.407	0.097	0.157	0.116	0.040	0.060
LexRank LR	0.313	0.213	0.253	0.074	0.050	0.060	0.288	0.196	0.233	0.083	0.082	0.083
LSA HR	0.435	0.113	0.179	0.130	0.032	0.052	0.401	0.104	0.166	0.114	0.043	0.063
LSA LR	0.270	0.250	**0.260**	0.050	0.048	0.049	0.245	0.227	0.235	0.071	0.097	0.082
LC HR	0.190	0.312	0.236	0.051	**0.078**	**0.062**	0.175	0.297	0.220	0.051	0.188	0.080
LC LR	0.381	0.143	0.208	0.111	0.039	0.057	0.351	0.133	0.193	0.100	0.062	0.077

TABLE (I) ROUGE-1, 2, L and W in dataset PS5K. LR corresponds to a lower ratio, and HR corresponds to a higher ratio.

Method	ROUGE-1			ROUGE-2			ROUGE-L			ROUGE-W		
	R	P	F	R	P	F	R	P	F	R	P	F
TF-IDF	0.224	0.175	**0.196**	0.023	0.018	0.020	0.206	0.161	0.181	0.059	0.076	**0.066**
TextRank HR	0.295	0.102	0.152	**0.041**	0.014	0.021	0.274	0.095	0.141	0.076	0.043	0.055
TextRank LR	0.187	0.197	0.192	0.017	0.019	0.018	0.171	0.183	0.177	0.050	0.088	0.064
QueSTS HR	0.187	0.201	0.194	0.024	0.020	**0.022**	0.175	0.189	**0.182**	0.051	0.093	**0.066**
QueSTS LR	0.097	**0.275**	0.143	0.009	**0.023**	0.013	0.090	**0.259**	0.134	0.029	**0.142**	0.048
Lexrank HR	**0.297**	0.102	0.152	0.040	0.014	0.021	**0.276**	0.095	0.141	**0.077**	0.043	0.055
Lexrank LR	0.196	0.194	0.195	0.019	0.020	0.019	0.182	0.181	0.181	0.053	0.087	**0.066**
LSA HR	0.296	0.114	0.164	0.036	0.013	0.019	0.273	0.105	0.151	0.076	0.048	0.059
LSA LR	0.164	0.212	0.185	0.011	0.016	0.013	0.149	0.194	0.168	0.044	0.094	0.060
LC HR	0.174	0.191	0.182	0.021	0.020	0.020	0.161	0.179	0.170	0.045	0.106	0.064
LC LR	0.276	0.118	0.165	0.036	0.014	0.020	0.256	0.110	0.153	0.072	0.051	0.060

TABLE (II) ROUGE-1, 2, L, W in dataset SciDuet. LR corresponds to a lower ratio, and HR corresponds to a higher ratio.

ROUGE-W has its best results with TF-IDF, QueSTS HR and LexRank LR.

However, looking at the best scores in related work [18] using the same dataset, the best reported ROUGE-1, ROUGE-2, and ROUGE-L scores are respectively 0.20, 0.05, and 0.19. For ROUGE-1, methods such as TF-IDF (0.196), LexRank LR (0.195) and QueSTS (0.194) perform very close to the state-of-the-art (0.20). For ROUGE-2, every method outperforms the best values obtained in the state-of-the-art. For ROUGE-L, QueSTS HR (0.182), LexRank LR (0.181), and TF-IDF (0.181) are very close to the best obtained (0.19).

C. Main conclusions

It is not an easy task to determine which is the best method after the experiences presented in the sections above. This happens because some methods do better than others in certain metrics, and we cannot be certain which one is more relevant. In the dataset PS5K, the best performing method seems to be TextRank, since it is the method that performs best in most metrics. It is loosely followed by: LSA LR, LexRank LR, QueSTS HR, LC HR, and TF-IDF. In the dataset SciDuet, the best approaches are TF-IDF and QueSTS (their values are all close), followed by LexRank LR, TextRank LR, LC HR, and LSA LR.

As we can see, the best and worst performing methods change a lot depending on the dataset, so choosing a method that is generally the best is not possible. Because of that, we had to try to determine which methods were, on average, better. LC is in both one of the worst, so it was excluded. TF-IDF is the best and the worst in different datasets, so it is also not a reliable choice. This is also similar for LSA, which is the second best in PS5k and the worst in SciDuet. That left us with QueSTS, LexRank, and TexRank, which are all graph-based and have quite similar results. Between the three, the best to worst order in PS5K is TextRank, LexRank, and QueSTS, and in SciDuet it is QueSTS, LexRank, and TextRank. So, we chose LexRank to continue our experiments, namely, slide generation, since it is the one always in the middle, not the worst, not the best.

V. Slide Generation

To complete the pipeline of slide generation, the sentences of the generated summaries need to be organised into slides. While a more complex strategy could be devised [13], [18], and will be, in the future, we adopted a fairly simple one for illustrating the process in this paper. Having in mind that most textual documents contain sections, we rely on their titles for defining slide topics. The sentences in the summary are then grouped according to those topics, i.e., each sentence is associated to the topic (section) under where it was placed, in the original document. As a result, each slide will feature a title (i.e., the title of the section) followed by sentences selected from the original document, in their original order. Each slide can only contain a configurable number of sentences. If the text does not fit in a single slide, additional text-only slides are made.

The datasets used on this paper are based on scientific research and therefore present a lot of complex terms and formulas. So, in order to present an example that everyone can easily understand a simpler text was extracted. For this, several articles from Wikipedia were chosen, all of which dealt with topics that a large number of people are likely to be familiar with. This will also make human evaluation easier because even if people do not know a lot about the topic, they can quickly read and understand the original article, which is impossible to do with scientific articles that require a certain level of expertise on the subject.

Experiments were made in both English and Portuguese, and as expected the method was able to perform in both languages.

For illustrative purposes, Figure 1 shows the slides generated with LexRank LR for the article "Europe", in the English Wikipedia.

It is important to remember that these slide decks are not the final product, but merely an initial draft that can be later improved. Humans can edit the slides in order to remove or add some sentences; add figures and tables; rectify some grammar and coherence problems in the text; etc.

VI. Conclusion

The objective of this work is to search for ways of creating an initial draft of slide decks based on a document. This will still require a human to review and edit the slides, but it will provide a starting point and thus speed up the process of creating slide decks. This will allow teachers who use slides on a regular basis to be more productive and spend their time on other projects.

In order to create the slide decks, we started by exploring a variety of unsupervised summarization methods for slide generation, with the goal of determining whether they are viable alternatives for this purpose. Several methods of this kind were tested in two datasets of scientific papers and their associated slides, PS5K and SciDuet. Resulting summaries were evaluated with the ROUGE metric and, when possible, results were compared with those reported for supervised approaches in the same datasets. In both datasets, ROUGE scores were in line with the state-of-the-art, which was outperformed in a minority of metrics and methods. These are positive results for approaches that do not require any training, suggesting that they can be viable alternatives, especially when enough data is not available, which is the case of slide generation for domains other than scientific papers. The same applies for languages other than English, because, to some extent, the unsupervised methods are language-independent, even if some require an external corpus for computing IDF scores.

In the future, the reported summarization experiments will be reproduced for a range of supervised approaches, some for extractive summarization (e.g., SVR [12]), and others for abstractive (e.g., based on transformers [18]), including the approaches originally applied to PS5K and SciDuet.

As for slide generation, this report presents a simple solution, but that already shows a promising starting point. In future work, these slides will be improved on, not only in the text front, but also in the sense of adding figures and tables.

Furthermore, this study relied only on automatic evaluation metrics, which are still limited because it cannot capture certain properties of a good summary. This is even more clear for slide presentations. Even if more time-consuming and requiring more logistics, a better evaluation should be performed by humans, who should better assess properties like readability, coherency, conciseness, fluency, informativeness, or consistency.

Regarding our final goal, the next step will be to submit several slide decks to humans with easy subjects that they are familiar with, in order to evaluate not only the summaries but also the resulting slides.

Acknowledgements

This work was funded by: project SmartEDU (CENTRO-01-0247-FEDER-072620), co-financed by FEDER, through PT2020, and by the Regional Operational Programme Centro 2020; and national funds through the FCT – Foundation for Science and Technology, I.P., within the scope of the project CISUC – UID/CEC/00326/2020 and by the European Social Fund, through the Regional Operational Program Centro 2020.

References

[1] Ravindranath Chowdary, M. Sravanthi, and P. Kumar. A system for query specific coherent text multi-document summarization. *International Journal on Artificial Intelligence Tools*, 19:597–626, 10 2010. https://doi.org/10.1142/S0218213010000339 doi:10.1142/S0218213010000339.

[2] Scott C. Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the Association for Information Science and Technology*, 41:391–407, 1990.

[3] Günes Erkan and Dragomir R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, dec 2004.

[4] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, 1998.

[5] Som Gupta and S. K. Gupta. Abstractive summarization: An overview of the state of the art. *Expert Syst. Appl.*, 121:49–65, 2019.

[6] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

[7] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880. ACL, July 2020.

[8] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. ACL.

[9] Hans Peter Luhn. The automatic creation of literature abstracts. *IBM J. Res. Dev.*, 2:159–165, 1958.

[10] Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In *Proceedings of 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain, July 2004. ACL.

[11] Athar Sefid, Prasenjit Mitra, Jian Wu, and C. Lee Giles. Extractive research slide generation using windowed labeling ranking. *ArXiv*, abs/2106.03246, 2021.

[12] Athar Sefid, Jian Wu, Prasenjit Mitra, and C. Lee Giles. Automatic slide generation for scientific papers. In *SciKnow@K-CAP*, 2019.

[13] Athar Sefid, Jian Wu, Prasenjit Mitra, and C. Lee Giles. Extractive research slide generation using windowed labeling ranking. *CoRR*, abs/2106.03246, 2021.

[14] Prakhar Sethi, Sameer S. Sonawane, Saumitra Khanwalker, and Ravindra B. Keskar. Automatic text summarization of news articles. *2017 International Conference on Big Data, IoT and Data Science (BID)*, pages 23–29, 2017.

[15] M. Sravanthi, C. Ravindranath Chowdary, and P. Sreenivasa Kumar. Quests: A query specific text summarization system. In *FLAIRS Conference*, 2008.

[16] M. Sravanthi, C. Ravindranath Chowdary, and P. Sreenivasa Kumar. Slidesgen: Automatic generation of presentation slides for a technical paper using summarization. In *FLAIRS Conference*, 2009.

[17] Josef Steinberger and Karel Jezek. Using latent semantic analysis in text summarization and summary evaluation. volume 4, 2004.

[18] Edward Sun, Yufeng Hou, Dakuo Wang, Yunfeng Zhang, and Nancy Xin Ru Wang. D2S: document-to-slide generation via query-based text summarization. 2021.

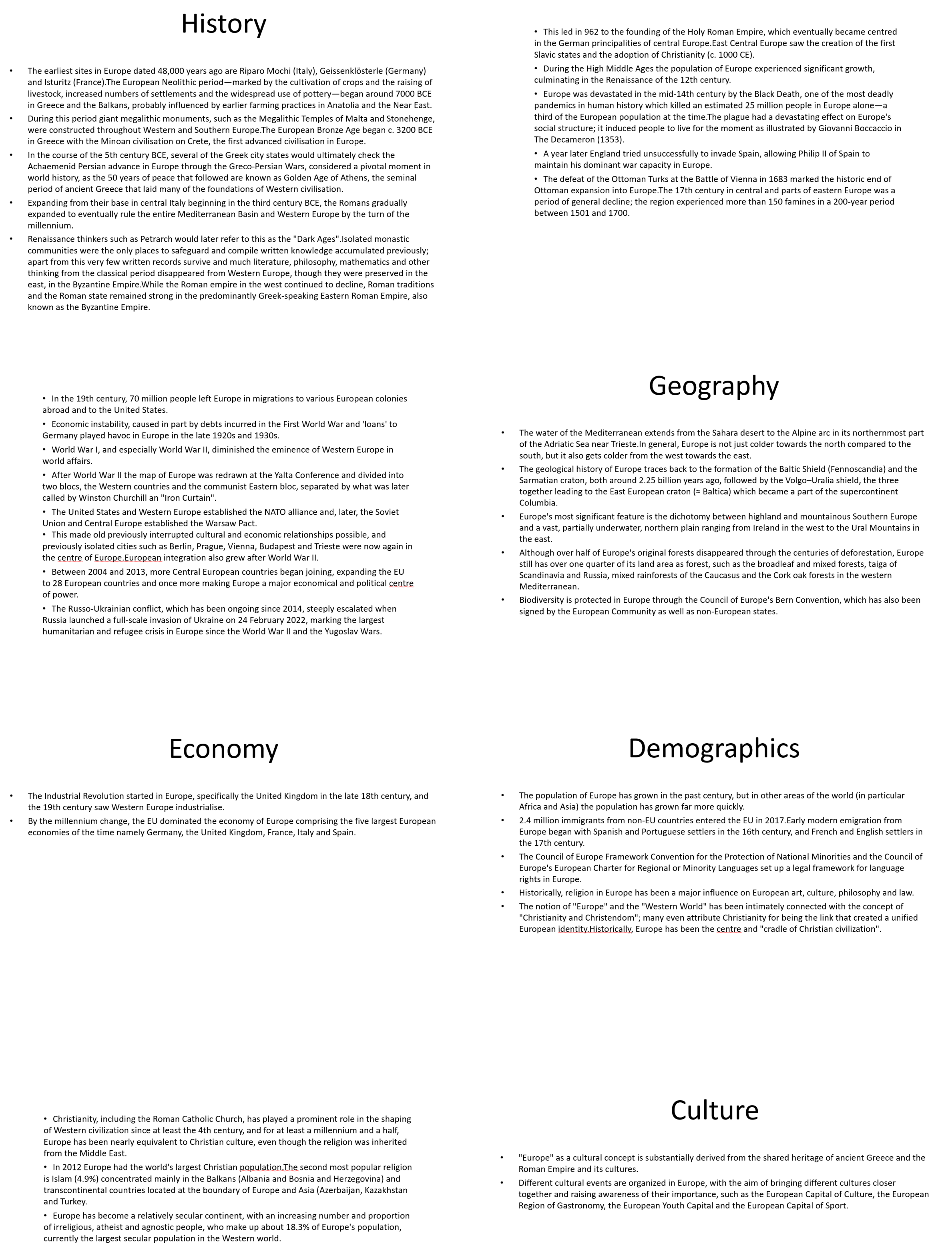

Fig. (1) Slides generated with LexRank for the article "Europe", in the English Wikipedia, as of June, 19, 2022. Read from left to right.