Proceedings of the 9th World Congress on Electrical Engineering and Computer Systems and Sciences (EECSS'23)
Brunel University, London, United Kingdom - August 03-05, 2023
Paper No. CIST 152
DOI: 10.11159/cist23.152

Historical-Domain Pre-trained Language Model for Historical Extractive Text Summarization

Lamsiyah Salima[1], Murugaraj Keerthana[1], Schommer Christoph[1]
[1] Department of Computer Science, Faculty of Science, Technology and Medicine, University of Luxembourg
4365 Esch-Belval Esch-sur-Alzette, Luxembourg
{firstname.lastname}@uni.lu

Abstract – In recent years, pre-trained language models (PLMs) have shown remarkable advancements in the extractive summarization task across diverse domains. However, there remains a lack of research specifically in the historical domain. In this paper, we propose a novel method for extractive historical single-document summarization that leverages the potential of a domain-aware historical bidirectional language model, pre-trained on a large-scale historical corpus. Subsequently, we fine-tune the language model specifically for the task of extractive historical single-document summarization. One major challenge for this task is the lack of annotated datasets for historical summarization. To address this issue, we construct a dataset by collecting archived historical documents from the Centre Virtuel de la Connaissance sur l'Europe (CVCE) group at the University of Luxembourg. Furthermore, to better learn the structural features of the input documents, we use a sentence position embedding mechanism that enables the model to learn the position information of sentences. The overall experimental results on our historical dataset collected from the CVCE group show that our method outperforms recent state-of-the-art methods in terms of ROUGE-1, ROUGE-2, and ROUGE-L F1 scores. To the best of our knowledge, this is the first work on extractive historical text summarization.

***Keywords*:** Extractive Text Summarization, Historical Domain, Pre-trained Language Models, HistBERT, Transfer Learning

1. Introduction

In recent decades, digital humanities have witnessed a substantial effort to digitize historical documents, leading to an unprecedented volume of machine-readable texts in digital format, often referred to as the "big data of the past" [1]. This abundance of historical information on the web and other sources highlights the need for extensive research in historical text processing. Such research is essential for analyzing and extracting valuable insights from this vast data, which can help historians and other users save time by quickly finding relevant information that meets their needs. Automatic Text Summarization (ATS) can be one of the effective tools to deal with this information overload in the historical domain.

Automatic Text Summarization (ATS) is the process of condensing a document or collection of documents while preserving their essential elements [2]. Generally, ATS is based on two commonly used approaches: ***extractive*** and ***abstractive*** summarization. *Extractive methods* involve identifying and extracting the most relevant sentences from the source documents without any other modification [3]–[5]. Abstractive methods aim to generate summaries with novel words or phrases based on natural language generation models [6], [7]. Furthermore, various types of automatic text summarization systems exist. These include single and multi-document summarization, as well as generic and query-focused summarization. Generic summaries cover all important details from a source document without considering specific user needs [8], while query-focused summaries are tailored to a user's information need or query [9]. As part of our research, we mainly focus on generic historical single-document summarization using an extractive approach.

Extractive text summarization methods generally involve three main steps: document analysis and representation, sentence scoring, and sentence selection. Traditional statistical or graph-based methods consider a document as a collection of sentences without capturing its global semantics [4], [10]. However, the emergence of deep learning, particularly pretrained language models, has revolutionized the extractive summarization approach showing promising results [5], [11]. These models can automatically learn textual features and effectively grasp the contextual information in the input documents. Generally, Deep learning-based extractive methods can be classified as supervised [11] or unsupervised [5]. Supervised extractive summarization is often formulated as a binary classification task that aims to determine whether a sentence should be included in the summary or not. In our research, we introduce a supervised extractive method for historical text summarization. However, a significant challenge we face is the limited availability of annotated dataset to train our proposed method.

To overcome the mentioned challenges, we initially create a historical dataset for single-document summarization. This involves gathering archived historical documents from the CVCE group at the University of Luxembourg[1]. Then, we identify relevant documents, perform data cleaning and pre-processing, and establish the master dataset. Inspired by the success of pre-trained language models (PLMs) in several natural language processing tasks [12], [13], including text summarization [11], [14], [15], we explore the potential of HistBERT model [16]. HistBERT is a domain-specific Language representation model that has been pre-trained on the large Corpus of Historical American English (COHA). In this work, we leverage the HistBERT encoder to capture both the token-level and the sentence-level context in the input documents and generate the sentence representations. Additionally, we introduce a sentence position embedding mechanism to learn the position information of sentences and better capture the structural information of the document. It is worth noting that the quality of the input representation significantly affects the output summary's quality. Our extractive model is then constructed on top of this encoder, incorporating multiple inter-sentence Transformer layers [17] to capture document-level features for sentence extraction.

We conducted several experiments on the constructed historical dataset to evaluate the effectiveness of the proposed method. The results demonstrate that our approach outperforms both traditional methods and recent state-of-the-art extractive deep learning models in terms of ROUGE-1, ROUGE-2, and ROUGE-L F1 scores. To the best of our knowledge, this is the first work for extractive historical text summarization using the HistBERT pre-trained language model.

The remainder of this paper is organized as follows: Section 2 presents a discussion on related work, Section 3 provides a detailed explanation of our proposed method, Section 4 describes and analyses the experimental results, and finally, Section 5 concludes the paper and draws lines for future research.

2. Related Work

In this work, we propose an extractive supervised method for historical single-document summarization based on the pre-trained HistBERT model [16]. Therefore, we will review research on historical NLP applications and extractive supervised text summarization methods. For a detailed review, readers may refer to [18], [19] and [20], [21] surveys, respectively.

2.1. Historical Natural Language Processing Applications

The definition of a historical document is not clearly defined and varies depending on different factors mentioned in existing literature, such as time, digital origin, type of writing, state of the material, and language. However, in [22], the author defines a historical document as any primarily textual document that was created or published before 1979, without considering its content, style, or how it was obtained. The year 1979 was chosen as a cut-off point because it is widely recognized as a significant turning point in history.

Existing NLP studies on historical documents primarily focus on tasks such as spelling normalization [18], [23], machine translation [24], and sequence labelling, including part-of-speech tagging [25] and named entity recognition [19], [26]. Recently, the success of deep neural networks has introduced new applications in this domain, including sentiment analysis [27], information retrieval [28], event extraction [29], [30], and text classification [31]. However, only a limited amount of research has been conducted on historical text summarization. In this context, Mahajan et al. [32] have proposed an extractive rule-based summarizer model for historical documents, which extracts the most important sentences from the input document based on some features like named entity, numerical data, and title word. In the same context, Peng et al. [33] have proposed a novel method for summarizing historical texts in their corresponding modern language. The authors experimented with the hundred years old standard text summarization

[1] https://www.cvce.eu/en

dataset consisting of historical German and Chinese news with their summarized text in modern German or Chinese. Their summarization model uses cross-lingual transfer learning techniques and was trained without parallel data. In this work, we aim to enhance the quality of English historical text summarization by exploiting the potential of the HistBERT model.

2.2. Neural Extractive Text Summarization

Extractive summarization methods generate a summary by identifying and concatenating the most relevant sentences sentences from a document. In neural models, extractive summarization is formulated as a sentence classification task. A neural encoder is used to create representations of sentences, and a classifier predicts which sentences should be selected as part of the summary. In this context, several methods have been proposed in the literature. For instance, SUMMARUNNER [6] was one of the earliest neural approaches that utilized a Recurrent Neural Network-based encoder for extractive summarization. REFRESH [34] employed reinforcement learning techniques and globally optimized the ROUGE metric for training. SUMO [35] introduced the concept of structured attention to represent the document using a multi-root dependency tree, enabling a more accurate prediction of the output summary. NEUSUM [36] achieved state-of-the-art performance by jointly scoring and selecting sentences in the extractive summarization task. Recently, Liu et al. [11] have introduced the BERTSUM model, a single document summarization model, which involved a novel BERT-based document-level encoder to capture document semantics and obtain sentence-level document representations. In the same context, Zhong et al. [37] proposed a single-document ex- tractive summarization method called MATCHSUM, which employs a new technique of Text Matching. The extractive summarization task is formulated as a semantic text- matching problem, where the candidate summary is matched with the input document in the semantic space. Motivated by the success of neural extractive text summarization models in general domains, we aim to explore their potential in the historical domain.

3. Proposed Method

3.1. Embedding Representation

Embedding representation is an essential step for any natural language processing application. It involves encoding words or sentences into vectors that capture their semantics using, for instance, pre-trained language models (e.g., BERT, HistBERT). The pre-trained HistBERT model is trained to predict individual words rather than whole sentences, which makes it challenging to use for text summarization. Although HitBERT incorporates segmentation embeddings to represent different sentences, these embeddings are designed for sentence-pair inputs and are not suitable for encoding and manipulating multi-sentential inputs required in summarization. To address this issue, we introduce our proposed architecture called HistBERTSum-Ext for historical extractive text summarization. The process of our method is depicted in Figure 1.

To represent the input sentences, we use three different embedding layers: Sentence Token, Sentence Position, and Token Position. Formally, given a historical document d, we first use the Stanford CoreNLP toolkit [38] for sentence splitting. Thus, the document d is represented as a set of n sentences, denoted as $d = \{S_1, S_2, \ldots, S_n\}$, where each sentence S_i in d is then converted into tokens. We add two tokens **[CLS]** and **[SEP]** at the beginning and the end of each sentence, respectively. The **[CLS]** token is used to provide information about the sentence's features, while the **[SEP]** token helps the model understand the subsequent sentence. The *sentence token embedding* layer converts each token t_j in S_i into an embedded vector with fixed dimensions using the pre-trained HistBERT model. We also use *sentence position embeddings* to distinguish multiple sentences within a document. For S_i we assign an embedding E_A or E_B depending on whether i is odd or even. Finally, the *token position embedding* encodes the position of each word within its respective sentence [14]. Using this approach, we can learn document representations in a hierarchical manner. The lower layers of the model focus on understanding adjacent sentences, while the higher layers, along with self-attention, capture the overall meaning and connections between multiple sentences. This hierarchical approach helps us capture both the details within individual sentences and the larger context of the entire document.

It is worth mentioning that the original HistBERT model has a maximum limit of 512 for position embeddings, which restricts the length of the input text it can handle. To work with longer texts, we introduce additional position embeddings that are randomly initialized and adjusted along with other model parameters.

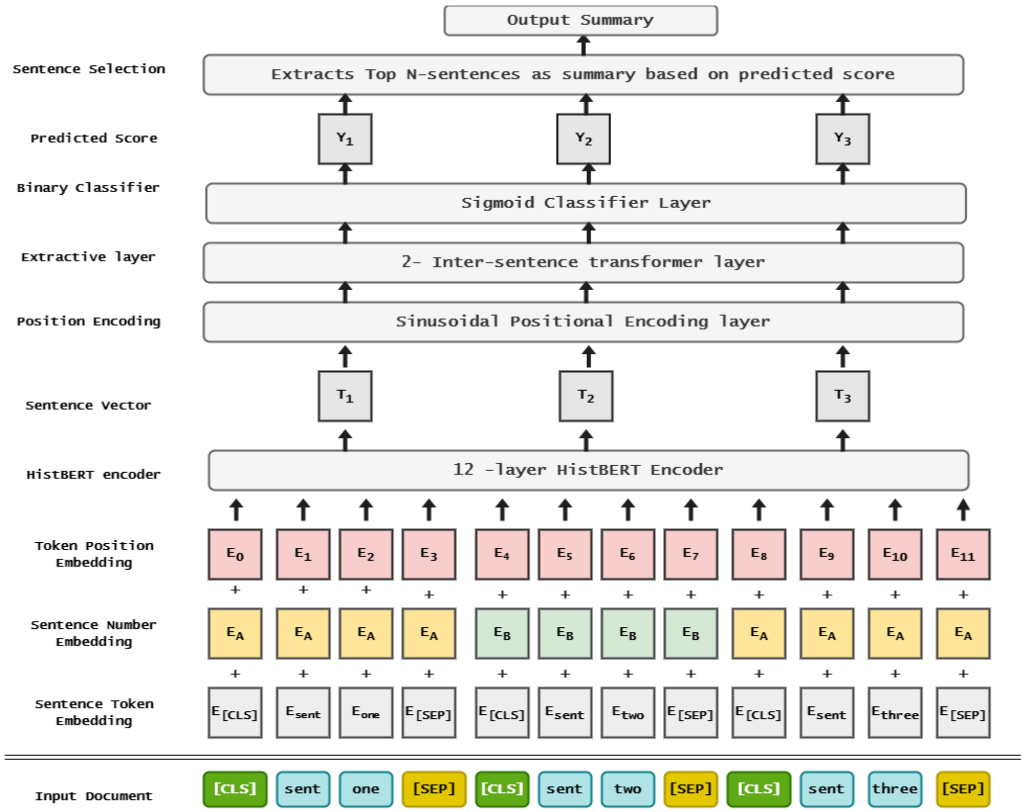

Fig. 1: HistBERTSUM-Ext Model Architecture. After adding two special tokens [CLS] and [SEP], the input document is passed to the three embedding layers. The final embedding representations are passed to the Encoder to obtain the sentence vectors. The document-level features are obtained from the extractive layer, and the sigmoid activation layer predicts the score of each sentence.

3.2. Extractive Summarization Process

Consider an input historical document d consisting of n sentences, denoted as $d = \{S_1, S_2, \ldots, S_n\}$, where S_i represents the *i-th* sentence in the document. Neural extractive summarization can be defined as the task of text classification where the model assigns a label $y_i \in \{0,1\}$ to each S_i, indicating whether the sentence should be included in the summary or not. The underlying assumption is that the sentences included in the summary represent the most important content of the document.

As shown in Figure 1, the top layer output of HistBERT produces vectors $T = \{T_1, T_2, \ldots, T_n\}$, where T_i is the vector of the *i-th* [CLS] symbol that is used as the representation for S_i. To capture the position of each sentence, we apply sinusoidal positional embedding to the sentence vectors $T = \{T_1, T_2, \ldots, T_n\}$ using Equation 1. Two inter-sentence transformer layers are then applied to these positional encoded sentence vectors T to capture document-level semantic information for summary extraction. These layers involve operations such as Layer Normalization (LN), Multi-Head Attention (MHAtt), and Feed-Forward Network (FFN), with the number of layers denoted by L=2. The output of the transformer layer, which represents document features, is passed through a sigmoid classification layer (Equation 4) to obtain the predicted scores $\tilde{y}_i$ for each sentence S_i. The top N sentences with the highest scores are selected as the final summary.

$$h^0 = PosEmb\ (T) \tag{1}$$

$$\tilde{h}_i^l = LN(h_i^{l-1} + MHAtt(h_i^{l-1})) \tag{2}$$

$$h_i^l = LN(\tilde{h}_i^l + FFN(\tilde{h}_i^l)) \quad (3)$$

$$\tilde{y}_i = \sigma(W h_i^l + b) \quad (4)$$

To train the model, we use the Binary Classification Entropy (BCE) loss between the gold labels of the sentences y_i and the predicted sentence scores $\tilde{y}_i$. The parameters of HistBERT and the summarization layers are fine-tuned together during training. We used Adam Optimizer (β_1= 0.9 and β_2= 0.999), and the learning rate (lr) is scheduled with 10,000 warmup steps: $lr = 2e^{-3}.\min(step^{-0.5}, step.warmup^{-1.5})$

4. Experimental Results

In this section, we first provide an overview of the dataset and evaluation measures used in our experiments. Then, we explain the experimental setup. Finally, we present and analyze the results we obtained.

4.1. Dataset and Evaluation Measures

To the best of our knowledge, there is currently no existing dataset for historical text summarization in English. To address this gap, we created our own dataset by collecting 7800 English documents from the Centre Virtuel de la Connaissance sur l'Europe (CVCE) group. The documents cover various topics, including news, politics, and interviews, and were written by CVCE experts. We downloaded the documents as PDFs and extracted relevant information such as document ID, title, abstract, page content, and number of pages. After thorough analysis and processing, we identified 3907 documents that are well-suited for our summarization task. To prepare the dataset for training, we divided these documents into three sets: a training set containing 3163 documents, a test set containing 372 documents, and a validation set also containing 372 documents.

For evaluation measure, we used ROUGE (Recall-Oriented Understudy for Gisting Evaluation [39]. Specifically, we have used ROUGE-N (ROUGE-1 and ROUGE-2) and ROUGE-L. ROUGE-N determines the similarity between the systems summaries and a set of gold summaries based on the n-gram overlap, whereas ROUGE-L evaluates the fluency of the summary, it is based on the Longest Common Subsequence (LCS) that considers the sentence level structure similarity. We have reported the obtained F1 performance of ROUGE-1 (R-1), ROUGE-2 (R-2), and ROUGE-L (R-L) using the official ROUGE toolkit (version 1.5.5) with standard options settings used for assessing extractive single-document summarization systems.

4.2. Experimental Setup

The proposed HistBERTSum-Ext method is implemented using PyTorch and based on the 'bert-base-uncased'[2] version of the HistBERT model, with BERT's subword tokenizer used for tokenization. It is trained on the high-performance Iris cluster[3] at the University of Luxembourg, which features 96 Nvidia V100 GPU-AI accelerators with Skylake or Broadwell processors. Specifically, we used 4 GPUs with ten cores and one node. Our models is trained for 70000 steps with gradient accumulation every two steps. Model checkpoints were saved, and the performance of the model was evaluated on the validation set every 1000 steps.

We used a greedy algorithm, like [6], to create an oracle summary for each document. This algorithm helps us select multiple sentences that give the highest possible ROUGE-2 score compared to the gold summary, which we then use to train our extractive models. When predicting a summary for a new document, we first employ the model to get the score for each sentence. Then, based on the obtained scores, we rank these sentences and select the top-3 ranked sentences as the final summary.

[2] https://github.com/wendyqiu/diachronicbert
[3] https://hpc-docs.uni.lu/systems/iris/

4.3. Comparison with State-of-the-art Methods

To assess the effectiveness of the proposed method, we compare its performance with other recent state-of-the-art methods for extractive historical single-document summarization. The obtained ROUGE F1 scores (R-1, R-2, and R-L) of our method HistBERTSum-Ext and the SOTA systems are presented in Table 1. The first block of the table depicts the ROUGE F1 scores of two extractive baselines, ORACLE and LEAD-3. ORACLE generates an extractive upper bound using a sentence selection technique that maximizes the ROUGE-2 score for the target, while LEAD-3 selects the first three sentences of the document as a summary. The second block of the table summarizes the F1 scores of several extractive SOTA systems using our dataset.

Table 1: R-1, R-2, R-L F1 score results of our method and SOTA extractive methods tested on our historical dataset.

System	R-1	R-2	R-L
ORACLE	33.60	15.81	24.94
LEAD-3	30.13	11.84	22.29
TextRank [10]	20.40	6.91	15.91
S-BERT [40]	25.23	8.98	18.04
MatchSum [37]	27.90	9.06	21.00
BERTSUM [11]	30.88	11.04	23.21
HistBERTSum-Ext (Ours)	**32.56**	**12.97**	**24.52**

According to the results shown in Table 1, our method **HistBERTSum-Ext** performed better than traditional methods like LEAD-3 and TextRank [10] across all evaluation measures. It also outperformed recent deep learning- based methods such as S-BERT [40], MatchSum [37], and Vanilla BERTSum [11] in terms of evaluation measures. Our method collectively outperforms all previously proposed extractive systems, only falling behind the ORACLE upper bound. These results indicate that using the HistBERT model, which was pre-trained on a domain-specific corpus called the historical COHA corpus[4] helped capture deep meaning and more accurate context of the source document, resulting in relevant summaries. These findings highlight the importance of fine-tuning pre-trained language models on domain-specific datasets to improve their performance.

5. Conclusion

In this paper, we proposed a novel method called HistBERTSum-Ext for extractive historical single-document summarization, which utilizes the HistBERT encoder - a domain-aware language model. We introduced an effective text representation method that consists of three embedding layers, including the sentence token, sentence position, and token position embeddings. Additionally, as no historical datasets exist, we collected documents from the Centre Virtuel de la Connaissance sur l'Europe (CVCE) group and created our dataset for historical text summarization. The comparison of several recent extractive text summarization systems on the collected dataset has proven the effectiveness of the proposed method. Specifically, the use of the HistBERT encoder has shown to be effective for the historical text summarization task. As far as we know, this is the first work for extractive historical text summarization in English, which could be a good starting point for future research in this field. In the future work, we plan to investigate the efficiency of pre-trained language models for *abstractive* historical text summarization.

Acknowledgements

We thank **Dr. Lars Wieneke** and his team from the Digital Infrastructure Department at Luxembourg Centre for Contemporary and Digital History, Luxembourg University, for providing the dataset used in this work.

[4] https://www.english-corpora.org/coha/

References

[1] Melissa M Terras. The rise of digitization. Digitisation perspectives, pages 3–20, 2011.

[2] Hans Peter Luhn. A statistical approach to mechanized encoding and searching of literary information. IBM Journal of research and development, 1(4):309–317, 1957.

[3] Vishal Gupta and Gurpreet Singh Lehal. A survey of text summarization extractive techniques. Journal of emerging technologies in web intelligence, 2(3):258–268, 2010.

[4] Rafael Ferreira, Luciano de Souza Cabral, Rafael Dueire Lins, Gabriel Pereira e Silva, Fred Freitas, George DC Cavalcanti, Rinaldo Lima, Steven J Simske, and Luciano Favaro. Assessing sentence scoring techniques for extractive text summarization. Expert systems with appliations, 40(14):5755–5764, 2013.

[5] Salima Lamsiyah, Abdelkader El Mahdaouy, Bernard Espinasse, and Saïd El Alaoui Ouatik. An unsupervised method for extractive multi-document summarization based on centroid approach and sentence embeddings. Expert Systems with Applications, 167:114152, 2021.

[6] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Ça˘glar Gulçehre, and Bing Xiang. 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. Association for Computational Linguistics, 2016.

[7] Hui Lin and Vincent Ng. Abstractive summarization: A survey of the state of the art. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 9815–9822, 2019.

[8] Yihong Gong and Xin Liu. Generic text summarization using relevance measure and latent semantic analysis. In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, pages 19–25, 2001.

[9] Seeger Fisher and Brian Roark. Query-focused summarization by supervised sentence ranking and skewed word distributions. In Proceedings of the Document Understanding Conference, DUC-2006, New York, USA, 2006.

[10] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In Proceedings of the 2004 conference on empirical methods in natural language processing, pages 404–411, 2004.

[11] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. 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 3730–3740.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, pages 4171– 4186.

[13] Praboda Rajapaksha, Reza Farahbakhsh, and Noel Crespi. Bert, xlnet or roberta: the best transfer learning model to detect clickbaits. IEEE Access, 9:154704–154716, 2021.

[14] Yongping Du, Qingxiao Li, Lulin Wang, and Yanqing He. Biomedicaldomain pre-trained language model for extractive summarization. Knowledge-Based Systems, 199:105964, 2020.

[15] 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, ranslation, and comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7871–7880.

[16] Wenjun Qiu and Yang Xu. Histbert: A pre-trained language model for diachronic lexical semantic analysis. arXiv preprint arXiv:2202.03612, 2022.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.

[18] Michael Piotrowski. Natural language processing for historical texts. Synthesis lectures on human language technologies, 5(2):1–157, 2012.

[19] Maud Ehrmann, Ahmed Hamdi, Elvys Linhares Pontes, Matteo Romanello, and Antoine Doucet. Named entity recognition and classification on historical documents: A survey. arXiv preprint arXiv:2109.11406, 2021.

[20] Muhammad F Mridha, Aklima Akter Lima, Kamruddin Nur, Sujoy Chandra Das, Mahmud Hasan, and Muhammad Mohsin Kabir. A survey of automatic text summarization: Progress, process and challenges. IEEE Access, 9:156043 156070, 2021.
[21] Wafaa S El-Kassas, Cherif R Salama, Ahmed A Rafea, and Hoda K Mohamed. Automatic text summarization: A comprehensive survey. Expert systems with applications, 165:113679, 2021.
[22] Frank Bösch. Zeitenwende 1979: Als die Welt von heute begann. CH Beck, 2019.
[23] Marcel Bollmann, Anders Søgaard, and Joachim Bingel. Multi-task learning for historical text normalization: Size matters. In Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP, pages 19–24, 2018.
[24] Erik Tjong Kim Sang, Marcel Bollmann, Remko Boschker, Francisco Casacuberta, Feike Dietz, Stefanie Dipper, Miguel Domingo, Rob van der Goot, Marjo van Koppen, Nikola Ljubeši, et al. The clin27 shared task: Translating historical text to contemporary language for improving automatic linguistic annotation. Computational Linguistics in the Netherlands Journal, 7:53–64, 2017.
[25] Yi Yang and Jacob Eisenstein. Part-of-speech tagging for historical English. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1318–1328.
[26] Baptiste Blouin, Benoit Favre, Jeremy Auguste, and Christian Henriot. Transferring modern named entity recognition to the historical domain: How to take the step? In Proceedings of the Workshop on Natural Language Processing for Digital Humanities, pages 152–162. NLP Association of India (NLPAI), 2021.
[27] William L. Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. Inducing domain-specific sentiment lexicons from unlabeled corpora. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 595–605, Austin, Texas, November 2016.
[28] Eva Pettersson, Jonas Lindström, Benny Jacobsson, and Rosemarie Fiebranz. Histsearch-implementation and evaluation of a web-based tool for automatic information extraction from historical text. In HistoInformatics@ DH, pages 25–36, 2016.
[29] Rachele Sprugnoli and Sara Tonelli. Novel event detection and classification for historical texts. Computational Linguistics, 45(2):229–265, 2019.
[30] Viet Dac Lai, Minh Van Nguyen, Heidi Kaufman, and Thien Huu Nguyen. Event extraction from historical texts: A new dataset for black rebellions. In Findings of the Association for Computational Linguistics: ACLIJCNLP 2021, pages 2390–2400, 2021.
[31] Shmuel Liebeskind and Chaya Liebeskind. Deep learning for period classification of historical hebrew texts. Journal of Data Mining & Digital Humanities, 2020, 2020.
[32] Partha Protim Ghosh, Rezvi Shahariar, and Muhammad Asif Hossain Khan. A rule based extractive text summarization technique for bangla news documents. International Journal of Modern Education and Computer Science, 10(12):44, 2018.
[33] Xutan Peng, Yi Zheng, Chenghua Lin, and Advaith Siddharthan. Summarising historical text in modern languages. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 3123–3142, April 2021.
[34] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Ranking sentences for extractive summarization with reinforcement learning. In Proceedings of the 2018 Conference of the North American Chapter of the Association or Computational Linguistics, pages 1747–1759, 2018.
[35] Yang Liu, Ivan Titov, and Mirella Lapata. Single document summarization as tree induction. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, pages 1745–1755, 2019.
[36] Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. Neural document summarization by jointly learning to score and select sentences. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 654–663, 2018.

[37] Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. Extractive summarization as text matching. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6197–6208, 2020.
[38] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, pages 55–60, 2014.
[39] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In Text Summarization Branches Out, pages 74–81. Association for Computational Linguistics, July 2004.
[40] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 3982–3992, 2019