Cross-lingual Cross-temporal Summarization: Dataset, Models, Evaluation

Ran Zhang*
Natural Language Learning Group
(NLLG), School of Business
Informatics and Mathematics,
University of Mannheim
ran.zhang@uni-mannheim.de

Jihed Ouni
Fachbereich Informatik,
Technische Universität Darmstadt
jihed.ouni@stud.tu-darmstadt.de

Steffen Eger
Natural Language Learning Group
(NLLG), School of Business
Informatics and Mathematics,
University of Mannheim
steffen.eger@uni-mannheim.de

While summarization has been extensively researched in natural language processing (NLP), cross-lingual cross-temporal summarization (CLCTS) is a largely unexplored area that has the potential to improve cross-cultural accessibility and understanding. This article comprehensively addresses the CLCTS task, including dataset creation, modeling, and evaluation. We (1) build the first CLCTS corpus with 328 instances for hDe-En (extended version with 455 instances) and 289 for hEn-De (extended version with 501 instances), leveraging historical fiction texts and Wikipedia summaries in English and German; (2) examine the effectiveness of popular transformer end-to-end models with different intermediate fine-tuning tasks; (3) explore the potential of GPT-3.5 as a summarizer; and (4) report evaluations from humans, GPT-4, and several recent automatic evaluation metrics. Our results indicate that intermediate task fine-tuned end-to-end models generate bad to moderate quality summaries while GPT-3.5, as a zero-shot summarizer, provides moderate to good quality outputs. GPT-3.5 also seems very adept at normalizing historical text. To assess data contamination in GPT-3.5, we design an adversarial attack scheme in which we find that GPT-3.5 performs slightly worse for unseen source documents compared to seen documents. Moreover, it sometimes hallucinates when the

[*] Corresponding author.

Action Editor: Mohit Bansal. Submission received: 13 July 2023; revised version received: 2 February 2024; accepted for publication: 30 April 2024.

https://doi.org/10.1162/coli_a_00519

source sentences are inverted against its prior knowledge with a summarization accuracy of 0.67 for plot omission, 0.71 for entity swap, and 0.53 for plot negation. Overall, our regression results of model performances suggest that longer, older, and more complex source texts (all of which are more characteristic for historical language variants) are harder to summarize for all models, indicating the difficulty of the CLCTS task. Regarding evaluation, we observe that both the GPT-4 and BERTScore correlate moderately with human evaluations, implicating great potential for future improvement.

1. Introduction

Summarization is a key task in natural language processing (NLP), especially in an age of information overload. The classical approach for summarization is to summarize documents (e.g., news reports) in the same language (Zhang et al. 2020a; Liu et al. 2022; Ravaut, Joty, and Chen 2022) or, less prominently, in different languages than the source (**cross-lingual summarization**; CLS) (Ladhak et al. 2020; Cao, Liu, and Wan 2020; Bai, Gao, and Huang 2021; Liang et al. 2022). Tasks such as the summarization of historical texts to modern languages (**cross-temporal summarization**; CTS) are scarcely explored, in contrast. Such summarizers are beneficial not only for historical researchers and students but also for laypeople with an interest in historical information sources.

In this work, we go one step further and explore **cross-lingual cross-temporal summarization (CLCTS)**, in which the goal is to summarize a historical document in a different modern language. See Table 1 for an example. Such cross-lingual and cross-temporal summarizers would potentially tremendously facilitate cross-cultural accessibility, information sharing, and understanding. CLCTS is a valuable area of research that expands upon the recently popular task of literary translation with summarization. Both tasks share cultural importance and present unique complexities due to the intricate nature of the creative work involved (Karpinska and Iyyer 2023; Thai et al. 2022). CLCTS is not limited to literary works alone: It can also assist in summarizing non-literature such as historical newspapers, reports, textbooks, and so forth. For example, CLCTS models can assist in creating (missing) Wikipedia pages for historical works. While the translation of literature may require language that is faithful to the original source for a "close reader." CLCTS could potentially be better suited for a "distant reader," and thus may require less sophisticated output text. Moreover, modern summaries can facilitate the classification of literary works, for example, genre classification (Saraswat and Srishti 2022; Agarwal, Vijay et al. 2021). Firstly, summaries are more condensed than (historical) source texts and, therefore, are much easier to model. Secondly, modern texts are better suited for models that operate with modern data such as SBERT (Reimers and Gurevych 2020).

CLCTS is difficult and challenging for the following reasons: (i) Languages change over time along multiple dimensions, including syntax (Juzek, Krielke, and Teich 2020; Lei and Wen 2020), semantics (Hamilton, Leskovec, and Jurafsky 2016; Giulianelli, Del Tredici, and Fernández 2020), and lexical choice and morphology (Gibson et al. 2019; Joseph 2017). (ii) Historical documents are often longer, not only because of tendencies of simplification/text length reduction over time (Lewis 1894; Sherman 1893; Rudnicka 2018), but also because of the genre of historical text, which predominantly includes literary work (Wang and Liu 2017; Zhu, Liu, and Pang 2022). (iii) The process can be thought of as consisting of several subtasks, previously considered independently in NLP research: machine translation (MT), summarization, and historical (spelling) normalization (Bollmann and Søgaard 2016; Eger, vor der Brück, and Mehler 2016).

Table 1
Example from our cross-lingual cross-temporal dataset (historical English to modern German). We highlight words with spelling/morphological changes in italics.

Title	An Inhabitant of Carcosa
Author	Ambrose Bierce
Year	1886
Source	The Collected Works of Ambrose Bierce, Volume 3
Text	"For there be divers sorts of death—some wherein the body *remaineth*; and in some it *vanisheth* quite away with the spirit . . . In one kind of death the spirit also *dieth*, and this it *hath* been known to do while yet the body was in vigor for many years. [...]"
Summary (German)	"Ein Mann aus der Stadt Carcosa, der über die Worte des Philosophen Hali über die Natur des Todes nachdenkt, wandert durch eine ihm unbekannte Wildnis. [...]"

In this work, we consider the CLCTS problem comprehensively, including dataset creation, modeling, and evaluation.[1] We start with building the first CLCTS dataset. Our corpus contains stories and plays (as prime artefacts of historical cultures), which requires the models to handle long documents; we assemble modern summaries by querying Wikipedia.

In terms of modeling, we consider three approaches: (1) Extractive summarizers (Gu, Ash, and Hahnloser 2022) whose advantage is that they can in principle deal with unlimited input document lengths but which are limited in that they can only copy out sentences from the original source (which is a problem, e.g., when the source is first-person narrative but the summary should be third-person perspective). (2) Abstractive summarizers can avoid the issue above but typically have severe limitations in input lengths. Even models exclusively designed for long document summarization by utilizing efficient variants of attention mechanism such as Longformer Encoder-Decoder (LED) (Beltagy, Peters, and Cohan 2020) and Bigbird (Zaheer et al. 2020) limit the max input tokens to 16,384 and 4,096, respectively. (3) GPT-3 (Brown et al. 2020) and its extensions such as ChatGPT are also strong candidates for our task (Goyal, Li, and Durrett 2022a) as they have so spectacularly impacted research in a multitude of scenarios (Leiter et al. 2024), including summarization (Yang et al. 2023; Wang et al. 2023; Zhang, Liu, and Zhang 2023). Consequently, we also include ChatGPT (GPT-3.5 and GPT-4) in our analysis.

In terms of evaluation, we consider three evaluation methods: (1) automatic evaluation, where we report multiple evaluation metrics additional to the commonly used variants of ROUGE (Lin 2004) in summarization tasks (Liang et al. 2022; Bai, Gao, and Huang 2021; Cao, Liu, and Wan 2020; Peng et al. 2021) since the performance of individual metrics may vary across datasets, challenging their reliability (Bhandari et al. 2020; Fabbri et al. 2021); (2) human evaluation, which reflects the actual quality of summaries according to human judgments and functions as a source of reliability measurement for automatic evaluation metrics; and (3) ChatGPT evaluation, where we examine the potential of ChatGPT as an alternative to human annotators given the same instruction considering the high cost of human evaluation (Gao et al. 2023) and issues

[1] Our code and data are available at `https://github.com/zhangr2021/CLCTS`.

of reproducibility (Belz et al. 2021; Belz, Thomson, and Reiter 2023; Chen, Belouadi, and Eger 2022). Our contributions are:

- To our best knowledge, we build the first CLCTS corpus, leveraging historical fiction texts and Wikipedia summaries.
- We examine the effectiveness of popular transformer end-to-end (e2e) models with different tasks for intermediate task fine-tuning (Chang and Lu 2021).
- We explore how ChatGPT performs for CLCTS.
- We provide human, GPT-4, and automatic evaluation for the task, showing that our intermediate task fine-tuned e2e models generate bad to moderate quality summaries; GPT-3.5 as a summarizer provides moderate to good quality outputs (without any fine-tuning); GPT-4 as an evaluator correlates with human evaluations to a moderate level.

2. Related Work

Historical Text Normalization. Historical text normalization, the process of mapping non-standard word tokens to their modern standard forms, is an important technique for analyzing historical texts. Model designs include (1) corpus-based lexical substitution as an effective component for normalization systems (Rayson, Archer, and Smith 2005; Baron and Rayson 2008; Bollmann 2012); (2) rule-based models, such as phonological rules (Porta, Sancho, and Gómez 2013) and edit distance measures (Adesam, Ahlberg, and Bouma 2012; Bollmann 2012); (3) statistical models which aim to maximize the probability of contemporary word form given its historical form (Pettersson 2016); and (4) neural network (NN) models which utilize NN architectures such as encoder-decoder LSTMs (Bollmann, Bingel, and Søgaard 2017; Robertson and Goldwater 2018) or RNNs (Makarov and Clematide 2020). Worth noting is that nearly all existing historical normalizers operate on words in isolation, except for the semi-supervised contextual normalizer from Makarov and Clematide (2020). In our study, we find that ChatGPT (GPT-3.5) has the potential to serve as a (much better) context-aware historical normalizer.

Cross-temporal Natural Language Generation (NLG). Especially CTS is an underexplored topic in NLP. Peng et al. (2021) first introduce the task of historical text summarization (CTS).[2] They point out the scarcity of CTS datasets and build the HISTSUMM dataset containing 100 historical German and Chinese news documents with reference summaries written by linguistic experts in both languages. Our work differs along several dimensions from Peng et al. (2021); we mostly explore e2e models, work on very different datasets, and consider transformation across both time and language—for example, CLCTS instead of only CTS—and provide a much more comprehensive evaluation. A related cross-temporal task in NLG is literary machine translation (e.g., Thai et al. [2022] build a large multilingual paragraph-level literary machine translation dataset PAR3 including paragraphs from historical novels and short stories with

[2] In their study, they regard such historical text summarization as a special case of CLS. We use the term CTS to avoid confusion and highlight the differences between CLS and CTS tasks.

multiple human-written English translations and machine translations from Google Translate). Because we focus on various summarizations instead of translations and long documents instead of paragraphs, the translation dataset is not within the scope of our current study.

Cross-lingual Summarization. One natural solution for the CLS task is to utilize a pipeline framework following a translate-then-summarize (Yao, Wan, and Xiao 2015; Ouyang, Song, and McKeown 2019) or summarize-then-translate (Wan, Li, and Xiao 2010) paradigm. To tackle the problem of error propagation from pipeline methods, Zhu et al. (2019) propose an e2e CLS framework where they incorporate monolingual summarization (MLS) and MT simultaneously into the CLS training process. Following this work, many researchers use related tasks additional to the target task either simultaneously (multi-task learning) or consecutively (intermediate task fine-tuning) to train e2e CLS summarizers (Weller, Seppi, and Gardner 2022). Takase and Okazaki (2022) combine translation and MLS datasets simultaneously in training a single encoder-decoder model and additionally use a special token in the input (e.g., <Trans> and <Summary>) to define the task. Bai et al. (2022) also use a multi-task learning framework by interpreting MT as a special CLS task where no source content should be discarded. Ladhak et al. (2020) experiment with two-step fine-tuning where the model is first fine-tuned for MT and then CLS and this framework delivers better results than pipeline approaches. Similarly, He et al. (2023) incorporate intermediate task fine-tuning with multiple summarization datasets before fine-tuning on the target task. Considering the effectiveness of intermediate task fine-tuning especially in low-resource settings (Chang and Lu 2021), we train our e2e models using the same approach by first fine-tuning with different intermediate tasks such as MLS or CLS before our target CLCTS task.

Long Document Summarization. This is another area of our concern. Current approaches for long document summarization can be classified into three categories, namely, extractive (Cui and Hu 2021), abstractive (Beltagy, Peters, and Cohan 2020), and hybrid summarization approaches (Pilault et al. 2020). Among the proposed models, BART (Lewis et al. 2020) and PEGASUS (Zhang et al. 2020a) are most popular transformer-based pretrained models used for long document summarization for both supervised abstractive (Zaheer et al. 2020; Huang et al. 2021) and hybrid models (Gidiotis and Tsoumakas 2020; Manakul and Gales 2021). As pointed out by Koh et al. (2022), abstractive fine-tuned models built with a combination of pretrained large language models (LLMs) and efficient attention mechanisms are among the most competitive approaches for long document summarization, for example, BART with LED attention (input limits 16,384) (Beltagy, Peters, and Cohan 2020) and PEGASUS with BigBird attention (4,096).[3] We build our model with mBART (Tang et al. 2020) pretrained for multilingual MT tasks and combine with LED attention to enable efficient processing for long sequences. Additionally, we build another retrieve-then-summarize pipeline where we retrieve sentences with an extractive summarizer (Gu, Ash, and Hahnloser 2022) and then summarize cross-lingually using ChatGPT.

[3] The very recently proposed retrieval-based approach Unlimiformer (Bertsch et al. 2024) extends the input limitations of the aforementioned pretrained encoder-decoder transformers to practically unlimited input sequences at test time by offloading the cross-attention computation.

ChatGPT for Summarization. Despite ChatGPT's young age, there is already a wealth of research that explores it for various NLP tasks, including summarization (Yang et al. 2023) and summarization evaluation (Chiang and Lee 2023; Gao et al. 2023). Disagreement exists concerning ChatGPT's performance. Most studies focus on MLS tasks. Soni and Wade (2023) find that humans are unable to distinguish between summaries written by humans and those produced by ChatGPT. However, ChatGPT as an extractive summarizer is inferior to existing supervised systems according to ROUGE (Zhang, Liu, and Zhang 2023). Bang et al. (2023) also find that the fine-tuned BART outperforms zero-shot ChatGPT by a large margin for MLS and point out that ChatGPT, like other LLMs, tends to generate hallucinated information beyond the given knowledge (extrinsic hallucination). As for CLS, Wang et al. (2023) examine the ability of ChatGPT with prompt engineering and claim that it outperforms other models on the task. They also claim that ChatGPT can perform better with pipeline-like prompt lines in an interactive manner. It is also worth noticing that all works mentioned above utilize small samples of 50 to 100 random samples (out of 3,000 to 60k observations) from each dataset in their experiments.

Summarization Evaluation Metrics. Depending on the availability of reference summaries, one can utilize *reference-based* metrics (focusing on overlap measure [Lin 2004; Popović 2015], embedding similarity [Zhang et al. 2020b; Zhao et al. 2019], discourse coherence [Zhao, Strube, and Eger 2023], natural language inference [NLI] [Chen and Eger 2023], and text generation tasks such as question answering [QA] [Yuan, Neubig, and Liu 2021; Wang, Cho, and Lewis 2020]) or *reference-free* metrics (Gao, Zhao, and Eger 2020; Chen, Li, and King 2021; Liu, Jia, and Zhu 2022; Laban et al. 2022; Belouadi and Eger 2023). In terms of summarization evaluation for ChatGPT or other variants of GPT, the above-mentioned ChatGPT summarization studies all choose ROUGE and yet Peyrard (2019) point out the low correlation between ROUGE and human judgments when evaluating high-scoring summaries. Furthermore, the study from Goyal, Li, and Durrett (2022a) shows disagreements between humans and 16 evaluation metrics[4] when ranking prompt-based GPT-3 and fine-tuned models. To address this concern, we leverage multiple automatic evaluation metrics, especially more recent ones (e.g., MENLI and DiscoScore), as well as human evaluation. Because we focus on long documents, we do not explore reference-free metrics, due to the vast mismatch in length between source texts and summaries.

As for recent developments in evaluation with LLMs, the very recently proposed metric G-Eval (Liu et al. 2023) develops a reference-free evaluation framework based on GPT-4 utilizing both chain-of-thought and a form-filling paradigm. Other studies explore simpler scenarios where LLMs such as ChatGPT are queried with prompts containing task instructions as inputs and directly output the evaluation results. Shen et al. (2023) find that the evaluation capability of LLMs depends on the evaluated dimensions where ChatGPT is more effective at evaluating consistency. Gao et al. (2023) conduct experiments with the summarization evaluation datasets SummEval (Fabbri et al. 2021) and Newsroom (Grusky, Naaman, and Artzi 2018) and find that ChatGPT outperforms automatic evaluation metrics (ROUGE, BERTScore, BARTScore, MoverScore) on one of the datasets (SummEval). Chiang and Lee (2023) provide more comprehensive experiments with recent LLMs including ChatGPT and find that ChatGPT can not only rate

[4] Reference-based metrics: ROUGE variants, METEOR, BLEU, BERTScore, MoverScore, QAEval variants. Reference-free metrics: SUPERT, BLANC, QuestEval, QAFactEval, FactCC, DAE, SummaC.

like human experts but also provide explanations for its own decisions. However, they also point out that ChatGPT evaluation is prone to giving lower scores. In our work, we also include GPT-4 as an evaluator where we query evaluation directly via GPT-4.

3. Dataset

In this section, we introduce our CLCTS corpus and additional sources we utilize for our experiments. We build our own corpus which covers English and German in both directions. We utilize multiple monolingual and cross-lingual summarization datasets as additional resources during our intermediate task fine-tuning process.

3.1 CLCTS Corpus

Manually creating summaries is very time-consuming and requires expert knowledge (Peng et al. 2021). Thus, we use fairy tales, short stories, and plays where summaries are available in Wikipedia articles. In each language direction, English and German, the dataset contains summary pairs from *historical documents* in one language to *modern summaries* in the other language.

Dataset Collection. We collect historical documents mainly from three different sources.[5] The links to the sources are collected in Section 8.1 (Appendix).

- *Deutsches Textarchiv* (DTA; German text archive): This is a basic stock of German-language texts focusing on the early 16th to early 20th centuries.
- *Wikisource*: This is a multilingual online digital library of free-textual content containing historical books in text format and original format stored as images. Some characters are changed from the original historical format to modern digital formats. Proofreading is performed for the texts by users on Wikisource following the proofreading guide.
- *Project Gutenberg*: This is a cultural digital archive with over 60,000 digitized books, including historical texts. It focuses on older works with expired U.S. copyrights.

Our workflow for building the corpus consists of two stages: (1) meta-information collection & historical text extraction and (2) summary collection & translation. In stage (1), we collect meta-information of historical documents using the "Beautifulsoup" library. We extract information such as titles, Web links, authors, years, and documents together with information from Wikipedia lists of German fairy tales and English short stories. For documents presented in several sources, the oldest version or the version with historical spelling is selected. For stage (2), we utilize the collected information to match the historical documents with their corresponding summaries from Wikipedia. The historical spelling of titles is converted to modern spelling to match the titles in Wikipedia. Figure 1 illustrates the process of data collection. If no original human correspondent summaries are found in the other language, we translate the summaries

[5] *Wikisource* and *Project Gutenberg* contain both German and English sources, while *Deutsches Textarchiv* contains German sources only.

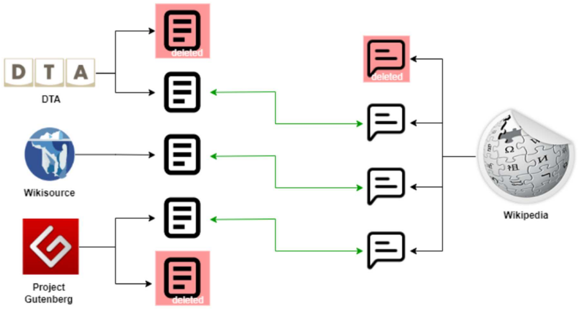

Figure 1
Texts matching from DTA, Wikisource, and Project Gutenberg to summaries collected from Wikipedia. Non-paired texts are excluded (highlighted in red).

into the target language using DeepL (we translate 144 out of 328 German summaries to English and 108 out of 289 English summaries to German).[6] An example of our collected dataset is shown in Table 1.

In addition to the summarization datasets, we create a historical German-English translation dataset with 201 translation pairs. This dataset is later utilized in intermediate task fine-tuning. We retrieve the German documents from the DTA corpus and the historical English from Wikisource. The documents are translated by Margaret Raine Hunt (1831–1912). An example of the translation dataset is given in Table 17 (Appendix).

We use the abbreviations hDe and hEn for historical German and historical English texts to distinguish them from modern texts De or En. Thus, we refer to the historical German to modern English summarization dataset as “hDe-En” and the historical English to modern German summarization dataset as “hEn-De.” Similarly, monolingual CTS datasets are called “hDe-De” or “hEn-En.” Names of other modern datasets follow the same rule.

3.2 External Datasets

To enrich our fine-tuning data resources, we utilize external summarization datasets: (1) HISTSUMM (Peng et al. 2021), where we also translate the summaries into English using DeepL to enable cross-lingual fine-tuning. (2) We further use Wikilingua for MLS and CLS tasks (Ladhak et al. 2020). Wikilingua is a multilingual summarization dataset that contains 18 languages from WikiHow (an online Wiki-style publication featuring how-to articles), including English and German. The English dataset contains 141,457 document-summary pairs. For fine-tuning our models, we randomly select 60% of the pairs (85,874) and we use all of the German dataset (58,341). Wikilingua German entries contain the URL and the section name of the English articles. Thus, cross-lingual summary pairs can be retrieved using this information and we match 20,103 cross-lingual summary pairs in both directions. (3) We also utilize the CNN/Daily Mail dataset. This is a monolingual English summarization dataset containing 311,971 document-summary pairs, including news articles written by CNN and Daily Mail journalists, and summaries from the highlights of the article written by the article author. We experiment with a subset of 86,133 text-summary pairs. (4) We include MLSUM (Scialom et al. 2020),

[6] DeepL API: `https://www.deepl.com/pro-api`.

Table 2
Characteristics of different datasets. We compute (1) the mean length of texts of both document and summary; (2) the mean length of sentences from documents (Sentence$_{\text{Doc.}}$) and sentences from summary (Sentence$_{\text{Summ.}}$). (3) Compression represents the document-summary level compression ratio (Mean Length Doc. divided by Mean Length Summ.). (4) Task represents the use case of the dataset during intermediate task fine-tuning: MLS, CLS, and CTS represent monolingual, cross-lingual, and cross-temporal summarization.

Dataset	**Size**	**Mean Length**				**Compression**	**Genre**	**Task**
		Doc.	Summ.	Sentence$_{\text{Doc.}}$	Sentence$_{\text{Summ.}}$			
				CLCTS Corpus				
CLCTS hEn-De	289	9,643.3	414.0	16.3	19.2	23.3	fiction	CLS, CTS
CLCTS hDe-En	328	1,398.4	324.3	25.4	18.6	4.3	fiction	CLS, CTS
CLCTS hEn-En	276	9,847.2	488.1	16.0	20.2	20.2	fiction	MLS, CTS
CLCTS hDe-De	328	1,398.4	217.4	25.4	16.6	6.4	fiction	MLS, CTS
				External Corpus				
HISTSUMM hDe-De	100	268.1	18.1	36.3	12.4	14.8	news	MLS, CTS
Wikilingua En-En	85,874	425.8	39.6	16.5	6.5	10.8	how-to-guide	MLS
Wikilingua De-De	58,341	429.8	42.2	15.8	6.9	10.2	how-to-guide	MLS
Wikilingua De-En	20,103	438.2	38.8	15.8	6.5	11.3	how-to-guide	CLS
Wikilingua En-De	20,103	451.7	42.7	16.5	6.9	10.6	how-to-guide	CLS
CNN/Daily Mail En-En	86,133	786.7	55.1	20.5	14.5	14.3	news	MLS
MLSUM De-De	66,226	570.3	30.4	17.8	12.7	18.8	news	MLS

which is a multilingual summarization dataset containing over 1.5 million document-summary pairs collected from online newspapers such as Süddeutsche Zeitung (German). We use a subset of 66,226 pairs from the German dataset.

3.3 Dataset Statistics

In Table 2, we report statistics of our corpus and other existing summarization datasets. We observe three trends comparing all datasets:

- *Much fewer instances in historical datasets.* The CLCTS hEn-De dataset consists of 289 historical short stories and plays, and the CLCTS hDe-En dataset consists of 328 historical fairy tales.[7] The size of HISTSUMM dataset is even smaller, with only 100 historical news articles. In contrast, modern summarization datasets contain several thousand summarization pairs, which is more than 100 times the size of historical datasets. The largest dataset CNN/Daily Mail contains over 300k summarization pairs.
- *Longer documents and summaries for the CLCTS corpus.* The CLCTS hEn-De dataset has more than 9,643 tokens per document and 414 tokens per summary on average and CLCTS hDe-En has 1,398 per document and 324 per summary. For external datasets, the average length is below 800 tokens per document and below 60 tokens per summary. This is because the CLCTS corpus includes historical documents from the fiction genre, which are naturally longer than news and WikiHow (how-to guide) due

[7] We limit the maximum length of the document to 16k. Since we collect summaries from Wikipedia, the total number of matched document-summary pairs is limited. Stories with less popularity are less likely to be found on Wikipedia than well-known ones.

Table 3
Divergence of sentence semantics measured by mean cosine similarity of sentence level embeddings between document and summary.

Monolingual		Cross-lingual	
CLCTS hDe-De	0.38	CLCTS hDe-En	0.33
CLCTS hEn-En	0.27	CLCTS hEn-De	0.25
HISTSUMM hDe-De	0.39	–	–
CNN/Daily Mail En-En	0.32	–	–
MLSUM De-De	0.37	–	–
Wikilingua De-De	0.46	Wikilingua De-En	0.42
Wikilingua En-En	0.42	Wikilingua En-De	0.42

to genre differences (Rudnicka 2018) and higher verbosity of more historical languages.

- *Larger information loss during summarization for news genre and CLCTS hEn-De measured by compression ratio.* News documents have a higher compression ratio (the mean length of document divided by the mean length of summary) compared with documents from other genres, except for CLCTS hEn-De, which has the highest compression ratio among all datasets. The CLCTS hEn-De dataset contains short stories and plays that are much longer than fairy tales in CLCTS hDe-En and documents from all other genres as discussed above. For the CLCTS corpus, the compression ratio differs substantially because all summaries of the CLCTS corpus are collected from Wikipedia with the average length of summaries ranging from 200 to 400 tokens regardless of the original document length.

To obtain more information on document-summary pairs, we quantify the divergence of sentence semantics between document and summary by computing the mean sentence-level cosine similarity for all datasets (Vyas, Niu, and Carpuat 2018). We obtain the sentence embedding via multilingual SBERT (Reimers and Gurevych 2020). We first compute the average similarity of each source to each summary sentence to obtain document level similarity and then average over all documents to obtain dataset level similarity score. The results are shown in Table 3. It reveals that cross-lingual sentence semantics between document and summary is more diverged than that of monolingual datasets and, among all, both CLCTS hDe-En and hEn-De (which are our target tasks) have lower embedding similarity compared to other datasets. This again indicates the difficulty of cross-lingual and cross-temporal summarization. We include an analysis of mean sentence length in the next section where we focus on a detailed analysis of historical documents and their comparison to modern documents.

3.4 Historical Language Divergence

Languages evolve due to social, cultural, and linguistic pressure; this poses challenges to the summarization of historical texts. In this section, we first provide an overview of the publication time of the historical documents from HISTSUMM and CLCTS, and second, analyze the divergence of historical language from syntax (including the length of sentences and dependency distances) and lexicon perspectives.

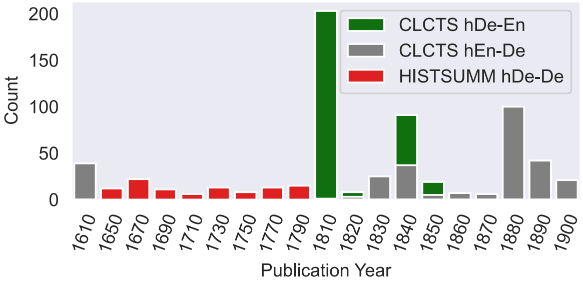

Figure 2
Distribution of publication year.

Publication Time of Historical Documents. The distribution of publication year is shown in Figure 2. HISTSUMM documents (in red) distribute evenly between the period of 1650–1800 CE. CLCTS hEn-De (in gray) contains more documents from the late 1800s and CLCTS hDe-En (in green) from the early and mid 1800s. The documents from CLCTS hEn-De, which date back to the 1600s, are plays from Shakespeare. For both CLCTS datasets, the majority of historical documents come from the 1800–1900s and comparatively HISTSUMM has overall older documents than our CLCTS corpus.

Divergence in Syntax. *Length of Sentences*: The analysis in Table 2 (column 5) reveals that historical German documents in HISTSUMM and CLCTS corpus contain longer sentences compared to other datasets, where HISTSUMM hDe-De has a mean sentence length of 36.3 tokens and CLCTS hDe-En/hDe-De dataset has 25.4 tokens per sentence while modern German documents from Wikilingua and MLSUM have maximum 20 tokens per sentence on average. This coincides with the finding of decreasing sentence length over time (Rudnicka 2018), considering that HISTSUMM documents are the oldest. Slightly surprisingly, this is not true for the English documents, where the mean sentence length of historical English documents (CLCTS hEn-De/hEn-En) is almost the same as Wikilingua and is even shorter than documents from the news genre CNN/Daily Mail. This hints at a genre effect, besides the temporal effect: Fiction, especially plays, may have shorter sentences than news and other non-fiction genres due to the higher occurrence of dialogues and two-word sentences (Rudnicka 2018). In our CLCTS corpus, unlike CLCTS hDe-En/hDe-De containing only fairy tales, historical English documents contain plays. When we exclude plays from Shakespeare, the mean sentence length of the CLCTS hEn-De document increases by 4 tokens per sentence to 20.3, which is closer to the value of CNN/Daily Mail. The factual summaries from CLCTS hEn-De/hEn-En in column 6 also contain longer sentences than their fictional documents for similar reasons.

Dependency Distances. To further quantify syntactic changes, we analyze the dependency parsing trees for sentences from documents in all datasets and compute the mean dependency distance (MDD) over the entire corpus (Liu 2008; Liu, Xu, and Liang 2017). This measure can reflect the syntactic complexity of sentences. We use stanza (Qi et al. 2020) for parsing the sentences. We plot the MDD in Figure 3 where we observe a downward trend overall. The first four documents from the left are in German and the last three documents are in English (we use dataset plus language, e.g., CLCTS hDe to represent the documents here). The downward trending line reveals an overall higher

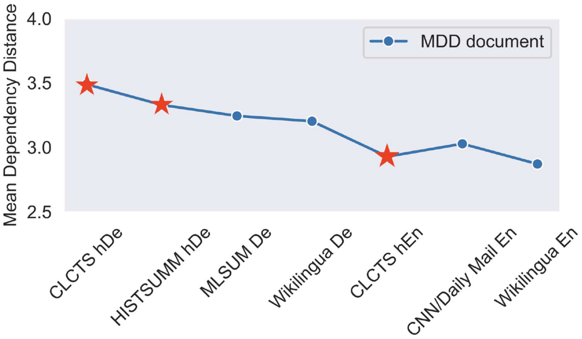

Figure 3
Mean dependency distance (MDD) for document sentences in each dataset. Historical documents are marked with red stars.

MDD of German sentences than that of English sentences. If we take a closer look by language, we observe: (1) For German (the first four points from the left), historical sentences (with a red star) have a higher value for MDD than that of the modern German text, which suggests more syntactic complexity of historical documents. We especially notice a higher MDD from CLCTS hDe documents (fiction) than that from HISTSUMM hDe (news), which coincides with the finding of longer dependencies in the imaginative genre than those in the informative genre from Wang and Liu (2017). The modern sentences from MLSUM are also more complex syntactically compared to Wikilingua. (2) For English (top-3 right items), this is not entirely true. News documents from CNN/Daily Mail have again a longer MDD than that of Wikilingua. However, they also have a longer MDD than CLCTS hEn. This may be because of the combined effect of (1) shorter sentences from CLCTS hEn (also from Wikilingua En) as discussed above, (2) more conversational texts with shorter dependencies from CLCTS hEn documents (plays and short stories) (Wang and Liu 2017), and (3) CLCTS hEn documents mostly from the late 1800s which share more similarities to modern English.

Divergence in Lexicon. The vocabulary used in historical and modern language varies greatly due to spelling, morphological, and semantic changes. For example, in our historical German dataset, the letter "β" often represents "ss" in modern texts (e.g., "daβ" vs. "dass"). Omitting "h", "t", or "e" is another type of spelling change such as "Thüre" (*door*, now "Türe" or "Tür"), "thun" (*to do*, now *tun*), "Todt" (*death*, now "Tod"), and "gieng" (passive tense of *to go*, now "ging"). Such changes in German are a result of two spelling reformations, one in 1901 and one in 1996. In English, we can also observe changes in lexicon such as the word "you" written as "thou" and the morphological change of the second person "will" as "wilt." To quantify these divergences, we compute the Jaccard similarity (Niwattanakul et al. 2013) of the lexicons between documents and their summaries in Table 4. Historical German has the lowest lexical overlap with modern German, with a Jaccard score of 0.1 for HISTSUMM and 0.19 for CLCTS hDe-De. The pattern is similar for English.

Semantic shift is another source of language change (Eger and Mehler 2016; Hamilton, Leskovec, and Jurafsky 2016). For example, the word "mistress" in Shakespeare's work has a similar meaning as "sweetheart." The "car" mentioned in Shakespeare's work (e.g., "Phoebus' car") refers to a special kind of chariot. Another example is the adjective "gay" which is used in the sense of "glad and cheerful," for example,

Table 4
Lexicon overlap measured by Jaccard score.

Dataset	Jaccard
CLCTS hEn-En	0.234
CNN/Daily Mail En-En	0.377
Wikilingua En-En	0.342
CLCTS hDe-De	0.186
HISTSUMM hDe-De	0.101
MLSUM De-De	0.335
Wikilingua De-De	0.314

"The great city in which he dwelt was very gay, for every day strangers visited the town" ("The Emperor's New Clothes" by Christian Andersen in 1888). Similarly, the meaning of the word "knave" degenerates from "servant" in the story of Brother Grimm to the modern meaning "a dishonest man" over time.

To conclude: (1) Historical German documents from HISTSUMM and CLCTS contain longer sentences with decreasing sentence length over time (temporal effect) while the sentence length of our historical English documents shows barely any difference to modern texts because of the genre effect. (2) German sentences show an overall higher MDD than that of English sentences in our entire corpus due to the joint effect of genre and publication time. (3) The vocabularies of historical and modern languages differ greatly in terms of spelling, morphological, and semantic changes.

4. Experimental Setup

In this section, we introduce the experimental setup of CLCTS. Figure 4 illustrates the model designs (detailed in Section 4.1) and evaluation strategies (detailed in Section 4.2).

4.1 Models

Our experiments consist of three methods: (1) pipeline methods based on extractive summarizers, (2) e2e transformer-based methods with intermediate task fine-tuning, and (3) zero-shot ChatGPT with different prompt strategies. For fine-tuned methods (extractive and abstractive), we use 5-fold cross-validation to obtain reliable results for all CLCTS datasets. For ChatGPT, we query summaries for all CLCTS datasets as well make sure the results are comparable.[8] We summarize model details in Table 5.

4.1.1 Pipeline Based on Extractive Summarizer. For extractive summarization, we first use an extract-then-translate pipeline. We choose MemSum (Gu, Ash, and Hahnloser 2022) as our model, which utilizes multi-step episodic Markov decision processes to extract important sentences from a given text. We use 5-fold cross-validation to train and test

[8] During 5-fold cross-validation, we accumulate the test sets from each fold and in this way, we obtain a final test set consisting of all samples. For example, for direction hDe-En with 328 samples, we accumulate output summaries from each test set consisting of 65-66 samples and overall we collect 328 samples after 5-fold cross-validation. In contrast, for zero-shot ChatGPT, we query directly from ChatGPT and collect 328 summaries at once.

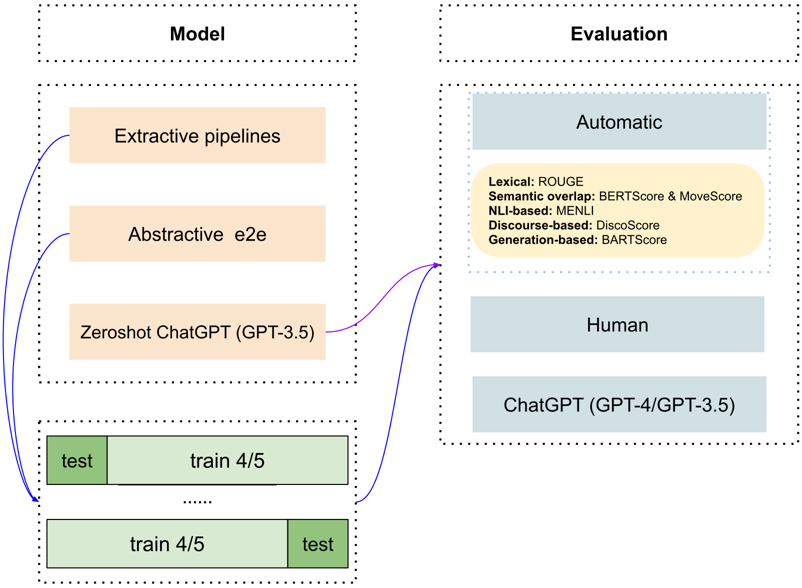

Figure 4
Flow chart illustrating models and evaluation strategies in Section 4.

Table 5
Summary of model designs. CV stands for 5-fold cross-validation.

Model name	Methods	Type	CV
MemSum			
-translation$_{\text{max 25}}$	extract-then-translate pipeline with 25 maximal extracted sentences per source document	extractive pipeline	✓
-Norma-translation$_{\text{max 25}}$	extract-normalize-translate pipeline with 25 maximal extracted sentences per source document		
mLED			
-Base	fine-tune with CLCTS target task alone		
-Historical MT	fine-tune with CLCTS historical translation dataset & CLCTS target task	abstractive e2e	✓
-Bidirection$_{\text{Prefix}}$	fine-tune with CLCTS non-target task & CLCTS target task		
mLED			
-MLS$_{\text{tgt}}$ -MLS$_{\text{src + tgt}}$ -MLS-CLS -MLS-CLS-CTS	fine-tune with intermediate tasks listed in Table 6 & CLCTS target task	abstractive e2e	✓
ChatGPT as summarizer			
-e2e (Title) prompt	prompt in e2e style using only Title and Author		
-e2e prompt	prompt in e2e style using source documents	zero-shot abstractive e2e	×
-pipeline prompt	prompt in pipeline style using source documents		
-retrieve-ChatGPT$_{\text{max 100}}$	retrieve max 100 key sentences & prompt in e2e style using retrieved texts	zero-shot abstractive e2e (two-step)	×

MemSum with our monolingual dataset. We limit the maximal extracted sentences per text to 25 according to the average length of summaries in Section 3. During inference, we sort the sentences according to the original text's order and then translate the extracted sentences to the target language using a multilingual machine translation model developed by Facebook (M2M) (Fan et al. 2021). We refer to this model as MemSum-translation$_{\text{max 25}}$. We additionally experiment with historical text normalization before translation. We use the normalization method Norma[9] (Bollmann 2012). We choose the combined version that leverages Lookup, Rule-based, and Distance-based normalization, which is claimed to perform well for both English and German (Bollmann 2019).[10] We refer to this model as MemSum-Norma-translation$_{\text{max 25}}$.

4.1.2 Transformer-based e2e Models. We modify pretrained models together with intermediate task fine-tuning to build e2e models. We propose a modified multilingual mBART50 model pretrained for machine translation (Tang et al. 2020) with an efficient attention mechanism taken from LED (Beltagy, Peters, and Cohan 2020) as model architecture. This architecture is suitable for multilingual long sequences as it benefits from the cross-linguality of mBART and the efficiency of processing long documents from LED. We refer to this model as mLED. We use 5-fold cross-validation to obtain reliable results. Due to resource limitations, our mLED models have a maximum of 4,096 input tokens and the number of tokens for the generated summary ranges from a minimum length of 100 tokens to a maximum length of 512 tokens.

Baselines. We first provide baselines of mLED fine-tuned only within the CLCTS corpus: (1) for baseline 1, we fine-tune the model only with the target task; for example, for the model in direction hDe-En, we only fine-tune mLED with our training set of the same direction. We refer to this model as **mLED-Base**. (2) For baseline 2 (baseline 1 + historical machine translation), we utilize our historical translation dataset. We first fine-tune with the translation task and then with the target task, which we call **mLED-Historical MT**. (3) Finally, for baseline 3, we fine-tune with datasets from *both* directions. We add prefixes to indicate the directions. We refer to this model as **mLED-Bidirection$_{\textbf{Prefix}}$**.

Intermediate Task Fine-tuning. Apart from the final fine-tuning step with our target CLCTS task, we conduct intermediate task fine-tuning using external sources including MLS, CLS, and CTS. Table 6 depicts the detailed design.

4.1.3 CLCTS with ChatGPT. We test the summarization ability of ChatGPT with our CLCTS corpus.

Prompt Strategies. We utilize GPT-3.5-turbo via the OpenAI API to query the summary. We set the temperature to 0.7 to allow for some randomness. We adopt (1) a pipeline translate-then-summarize prompt as well as (2) an e2e prompt. We set prompt lines to the same language as the target output since ChatGPT works better in this setting according to our initial experiments.[11] We truncate the texts that exceed ChatGPT input limits.[12] We also utilize other information such as title and author to test if ChatGPT has

[9] `https://github.com/comphist/norma`.

[10] Norma is recommended by the author when only little training data is available.

[11] During our first round of data collection all the prompt lines were in English. We noticed that ChatGPT tends to output summaries in the wrong target language, especially when the summary is expected to be in German.

[12] According to our experiment with the GPT2 tokenizer, we find that on average our German texts have 1.73 tokens per word (which is higher than the suggested 1.3 tokens per word ratio from OpenAI) and English 1.07 tokens per word. Therefore, we limit the German text to 2,048 words and English to 3,000 words. Since the 4,096 tokens limits from GPT-3.5-turbo are shared between both prompts and completions (outputs), we need to leave some tokens for the output summary.

Table 6
Intermediate task fine-tuning experiments for abstractive e2e models. Src and tgt represent monolingual summarization tasks corresponding to the source or target language of the target CLCTS task. The selected tasks and datasets are listed.

Model	**MLS**	**CLS**	**CTS**
MLS$_{tgt}$	CNN/Daily Mail En-En (tgt)		
MLS$_{src+tgt}$	MLSUM De-De (src) CNN/Daily Mail En-En (tgt)		
MLS-CLS	Wikilingua De-De & En-En	Wikilingua De-En	
MLS-CLS-CTS	Wikilingua De-De & En-En	Wikilingua De-En	HISTSUMM hDe-En

(1) Direction hDe-En

Model	**MLS**	**CLS**	**CTS**
MLS$_{tgt}$	MLSUM De-De (tgt)		
MLS$_{src+tgt}$	Wikilingua En-En (src) Wikilingua De-De (tgt)		
MLS-CLS	Wikilingua En-En & De-De	Wikilingua En-De	
MLS-CLS-CTS	Wikilingua En-En & De-De	Wikilingua En-De	HISTSUMM hDe-De

(2) Direction hEn-De

Table 7
Prompt lines of ChatGPT CLCTS.

Type	**Src-Tgt**	**Prompt**
e2e	hDe-En	Please summarize the following text in English: [Text].
e2e	hEn-De	Bitte fasse den folgenden Text auf Deutsch zusammen: [Text].
e2e (Title)	De-En	Please give me the summary of the story [Title] written by [Author].
e2e (Title)	En-De	Bitte gebe mir die Zusammenfassung der Geschichte [Titel] von [Autor].
pipeline	hDe-En	Please first translate the following text into English and summarize the translated text: [Text]
pipeline	hEn-De	Bitte übersetze zuerst den folgenden Text auf Deutsch und fasse den übersetzten Text zusammen: [Text].

explicit "prior" knowledge of the input document and its corresponding summary. We list all our prompt lines in Table 7.

Retrieve-then-summarize with ChatGPT. To tackle input limitations of ChatGPT (e.g., for GPT-3.5-turbo a total of 4,096 tokens are shared between prompt and completion),[13] we build the retrieve-then-summarize design, where we use MemSum as a *key sentence retriever* and input the retrieved sentences to ChatGPT and request summarization in the target language. For this experiment, we limit the maximal extracted sentences per document to 100 according to the input limits of ChatGPT.

4.2 Evaluation

To identify the best model, we utilize multiple recently popular reference-based metrics. We also conduct a multi-phase human evaluation to evaluate the model and test the effectiveness of the resulting metrics. During the multi-phase evaluation, we give

[13] https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them.

Table 8
Details of the evaluation strategies.

Type	Aspects	Source	Detail
Automatic	Lexical overlap	ROUGE-variant (Lin 2004)	*ROUGE-1* matching the texts through computing the unigram overlap; *ROUGE-L* leveraging the longest common subsequence
	Semantic overlap	BERTScore (Zhang et al. 2020b)	measuring semantic overlap through calculating the token similarity
		MoverScore (Zhao et al. 2019)	measuring semantic overlap utilizing Word Mover's distance
	Generation-based	BARTScore (Yuan, Neubig, and Liu 2021)	metric based on text generation using BART
	NLI-based	MENLI (Chen and Eger 2023)	metric based on natural language inference
	Discourse-based	DiscoScore (Zhao, Strube, and Eger 2023)	metric using BERT to model discourse coherence
Human	consistency, coherence, fluency, and relevance	SummEval (Fabbri et al. 2021)	6 annotators, 130 annotated instances for direction hDe-En; 104 annotated instances for hEn-De.
ChatGPT	prompting	–	prompts consisting of the same instruction for human annotators; 494 annotations for hDe-En & 407 for hEn-De.

feedback to annotators after each phase for them to better align with the instructions. We list the details of all three evaluation strategies in Table 8.

Automatic Evaluation. We consider six scores as automatic metrics to reflect five different aspects for measuring the quality of output summaries. The configurations used for each automatic metric are presented in Table 18 (Appendix).

Human Evaluation. We use six "expert" annotators to evaluate all models. Three of the annotators are female and three are male. All annotators have high competence in English as they are affiliated with the university: they are Master students (3), postgraduates (1), PhD students (1), and NLP faculty members (1). Three of them are native German speakers. Thus, the generated English summaries are evaluated by all six annotators and the generated German summaries are annotated by three native speakers.

We utilize the evaluation strategy by Fabbri et al. (2021), where the annotators score the summary from four perspectives: coherence, consistency, fluency, and relevance. The outputs are rated with a score ranging from 1 (worst) to 5 (best) for each dimension and we allow 0.5 increments. The annotators are presented with the source document, one reference summary, and one output summary during annotation.[14] After phase 1, we give feedback to the annotators by providing a comparison of their scores with the average scores. The scores that differ substantially are highlighted and we encourage them to review the differences. To speed up the process, we did not evaluate the same number of documents for all the models. Nonetheless, we make sure that each model is evaluated for at least 4 identical source documents. For hDe-En, we collect 130 summary level instances and 104 instances for hEn-De.

[14] The annotation details including instructions and the number of annotated instances per model are given in Section 8.5 (Appendix).

ChatGPT Evaluation. Since human annotation is costly, we follow related work (Gao et al. 2023; Chiang and Lee 2023) and test ChatGPT as an evaluator for our CLCTS task. We differ in the following aspects: (1) our CLCTS task is harder to evaluate due to the long length of both reference and generated summary. (2) Our CLCTS task involves multilingual evaluation for both English and German. We give the same instructions to ChatGPT as to the human annotators. For reproducibility purposes, we set the temperature to 0 to reduce the randomness. We collect evaluations for all available generated summaries from documents that have been annotated by humans in the previous step, resulting in 494 ChatGPT annotations for hDe-En and 407 ChatGPT annotations for hEn-De.

5. Results

5.1 Automatic Evaluation Metrics

The experiment results are shown in Table 9 for hDe-En (upper) and hEn-De (lower).

Supervised Extractive Pipelines. *Finding 1. Slightly diminishing results from historical text normalization*: We observe a decrease in all metrics scores on MemSum pipeline normalized with Norma for both hDe-En and hEn-De. This is contrary to our hypothesis that spelling normalization of historical texts can boost the translation quality for translators originally trained for modern text. Even though Norma performs well at token-level normalization, it fails to consider the context of the words, where in our use case some historical spellings may correspond to multiple modern meanings depending on the context. An example is the word "in" which can function as an adverb in the phrase, for example, "in den Wald" (in the forest) or the accusative case of the word "ihn" (him) in historical texts. A false normalization of such cases results in the meaning change of the entire sentence.

Supervised Abstractive e2e. *Finding 1. Mixed results on intermediate task fine-tuning*: For **hDe-En**, intermediate task fine-tuning with external sources does not improve scores, and sometimes fine-tuning with more tasks even deteriorates the outcome according to the metrics. This may be because our CLCTS hDe-En dataset, which dates back to the early 1800s, has substantial linguistic differences from the external sources as discussed in Section 3. For **hEn-De**, the results show the opposite. We observe higher scores for models trained with more intermediate summarization tasks from external sources. Though, according to MENLI variants and BARTScore, it is not always better to add CTS tasks. This may be because the CTS tasks from HISTSUMM contain documents that have up to 200 years of difference in publication time and again lead to substantial linguistic differences.

Finding 2. Historical translation task improves model performance only mildly: For **hDe-En**, one of the best fine-tuned models is mLED-Historical MT, where we fine-tune intermediately with our historical translation task before our target CLCTS task, though according to the metrics, the improvement is very small. Similarly, for the other direction **hEn-De**, the historical translation task improves model scores but only mildly compared to intermediate tasks. Since the historical translation dataset contains only 201 pairs, a larger size of the historical translation dataset may further improve the results.

Zero-shot abstractive ChatGPT. *Finding 1. A close match between pipeline and e2e prompts:* For **hDe-En**, we observe slightly better scores for summaries generated with e2e

Table 9
Experiment results for CLCTS hEn-De and hDe-En. The best two results among all models are in bold font. The best score for the supervised abstractive models is indicated with an underline. MENLI-W represents MENLI combined with BERTScore-F1 with different weights using formula MENLI-W$_i$ = i × NLI-D + (1-i) × BERTScore-F1. See Table 18 (Appendix) for more information on the metric configuration.

Model	ROUGE-1/L	BERTScore-P/R/F1	BARTScore	MoverScore	MENLI-W$_1$/W$_{.8}$/W$_{.3}$/W$_{.2}$	DiscoScore
Supervised Extractive						
MemSum						
-translation$_{\text{max 25}}$	0.320/0.175	0.521/0.574/0.545	−**3.311**	0.552	−0.232/0.407/0.463/0.474	0.551
-Norma-translation$_{\text{max 25}}$	0.315/0.171	0.514/0.567/0.538	−**3.360**	0.551	−0.248/0.398/0.450/0.461	0.536
Supervised Abstractive - Baseline CLCTS						
mLED						
-Base	0.391/0.201	0.537/0.561/0.547	−3.524	0.568	−0.206/0.418/0.469/0.479	**1.838**
-Historical MT	**0.393**/0.200	0.539/0.566/0.551	−3.492	0.569	−0.216/0.416/0.473/0.485	**1.725**
-Bidirection$_{\text{Prefix}}$	0.388/0.198	0.537/0.561/0.547	−3.534	0.567	−0.231/0.408/0.465/0.477	1.142
Supervised Abstractive - intermediate fine-tuning						
mLED						
-MLS$_{\text{tgt}}$	0.389/0.198	0.540/0.556/0.547	−3.567	0.568	−0.255/0.398/0.461/0.473	1.426
-MLS$_{\text{src + tgt}}$	0.386/0.201	0.536/0.547/0.54	−3.593	0.568	−0.252/0.397/0.453/0.464	1.595
-MLS-CLS	0.380/0.198	0.530/0.544/0.536	−3.615	0.567	−0.241/0.400/0.448/0.458	1.685
-MLS-CLS-CTS	0.386/0.202	0.532/0.548/0.539	−3.590	0.568	−0.246/0.399/0.451/0.462	1.635
Zero-shot Abstractive						
ChatGPT as summarizer						
-e2e (Title) prompt	0.304/0.164	0.527/0.530/0.528	−3.857	0.542	−0.700/0.211/0.367/0.398	0.788
-e2e prompt	**0.399**/**0.244**	**0.646**/**0.607**/**0.624**	−3.363	**0.575**	−**0.194**/**0.455**/**0.580**/**0.605**	0.982
-pipeline prompt	0.382/0.232	0.637/**0.597**/**0.615**	−3.422	**0.571**	−0.282/0.415/0.554/0.581	0.907
-retrieve-ChatGPT$_{\text{max 100}}$	0.382/**0.234**	**0.639**/**0.597**/**0.615**	−3.421	**0.571**	−**0.206**/**0.446**/**0.566**/**0.589**	0.883

(a) Direction hDe-En

Model	ROUGE-1/L	BERTScore-P/R/F1	BARTScore	MoverScore	MENLI-W_1/$W_{.8}$/$W_{.3}$/$W_{.2}$	DiscoScore
Supervised Extractive Pipeline						
MemSum						
-translation$_{\text{max 25}}$	0.315/0.131	0.830/0.839/0.834	−5.320	0.814	−0.648/0.236/0.387/0.418	0.301
-Norma-translation$_{\text{max 25}}$	0.311/0.130	0.829/0.838/0.833	−5.350	0.814	−0.652/0.234/0.383/0.413	0.291
Supervised Abstractive - Baseline CLCTS						
mLED						
-Base	0.321/0.142	0.853/0.843/0.848	−5.275	0.823	−0.605/0.273/0.462/0.499	**1.653**
-Historical MT	**0.328**/0.143	0.853/**0.845**/0.849	−5.301	0.823	−0.600/**0.276**/0.465/0.503	1.411
-Bidirection$_{\text{Prefix}}$	**0.328**/0.144	0.854/0.844/0.849	−5.276	0.824	−**0.601**/**0.276**/0.467/0.505	1.157
Supervised Abstractive - intermediate fine-tuning						
mLED						
-MLS$_{\text{tgt}}$	0.316/0.141	0.855/0.841/0.848	−5.253	**0.826**	−0.613/0.269/0.460/0.498	1.500
-MLS$_{\text{src+tgt}}$	0.321/0.145	0.852/0.841/0.847	−5.311	0.825	−0.618/0.266/0.453/0.491	**1.804**
-MLS-CLS	**0.328**/0.147	0.855/0.844/0.850	−5.250	**0.826**	−**0.586**/**0.283**/0.472/0.510	1.050
-MLS-CLS-CTS	**0.341**/**0.153**	0.855/**0.845**/0.850	−5.255	**0.827**	-0.624/0.268/0.468/0.508	0.764
Zero-shot Abstractive						
ChatGPT as summarizer						
-e2e (Title) prompt	0.288/0.135	0.857/0.840/0.849	−5.009	0.822	−0.745/0.218/0.444/0.489	0.590
-e2e prompt	0.282/**0.150**	**0.876**/0.843/**0.859**	−4.949	0.823	−0.655/0.269/**0.509**/**0.557**	0.572
-pipeline prompt	0.285/0.147	**0.874**/0.843/**0.858**	−**4.933**	0.824	−0.636/0.274/0.505/0.551	0.797
-retrieve-ChatGPT$_{\text{max 100}}$	0.269/0.143	**0.874**/0.843/**0.858**	−**4.914**	0.824	−0.638/0.274/**0.506**/**0.552**	0.605

(b) Direction hEn-De

prompts which tell ChatGPT to summarize in the target language directly (see Table 7). However, in Section 6, we show that the e2e prompts are prone to invalid outputs.

Finding 2. Competitive results from the retrieve-then-summarize design with ChatGPT: We observe competitive results from retrieve-then-summarize design for both hDe-En and hEn-De. Especially for hDe-En, it is scored as one of the top two models by ROUGE-L, MoverScore, BERTScore, and MENLI. Also worth noting is that this design not only performs well but also expands ChatGPT's potential for long document summarization.

Finding 3. ChatGPT can output summaries from its memory: For the e2e (Title) prompt experiment, we input zero text from the document but only the title, author, and publication year. Slightly surprisingly, for both directions, the resulting scores are mostly comparable to that of the supervised extractive models if not better. ChatGPT is capable of outputting summaries based solely on meta-information for all of our documents. This raises our question on the actual summarization ability of ChatGPT and therefore in Section 6, we design multiple tests to investigate in more detail about ChatGPT involving the CLCTS task.

Table 10
Annotation agreement measured by mean Spearman's correlation coefficient. Coh., Con., Flu, and Rel. represent coherence, consistency, fluency, and relevance, respectively.

Models	Coh.	Con.	Flu.	Rel.
hDe-En	0.497	0.595	0.150	0.572
hEn-De	0.353	0.493	0.433	0.605

Comparison Among All Approaches. All the metrics give lower scores to extract-then-translate models (with/without normalization) and the ChatGPT e2e (Title) prompt than other models, except for BARTScore, which rates the MemSum pipeline as the highest for hDE-En. However, *mixed results* are observed from the metrics for determining the best models. BERTScore variants in both directions give higher scores to ChatGPT-related models (excluding e2e (Title) prompt). MENLI variants agree with this fully for hDe-En and partially agree for hEn-De. The same is observed from MoverScore and ROUGE-L. BARTScore contradicts them and scores ChatGPT models higher for hEn-De, though not for hDe-En. Moreover, for hEn-De, ChatGPT is no longer dominating. However, the study from Goyal, Li, and Durrett (2022a) also shows that current metrics fail to evaluate the zero-shot model output properly. To better understand the performance of models and the metrics, we also consider human evaluation.

According to our automatic metrics, we find (1) extractive pipelines perform the worst among all methods; (2) supervised abstractive e2e methods outperform pipeline methods, where the historical translation task improves model performance mildly for both directions and we speculate that the effect of intermediate task fine-tuning depends on the linguistic similarity of external sources and target sources; and (3) zero-shot abstractive ChatGPT delivers the best results under e2e prompts but can output summaries from its memory based on the experiment results from e2e (Title) prompts.

5.2 Human and ChatGPT Evaluation

Table 10 shows the average Spearman's ranking correlation coefficient among the annotators for coherence, consistency, fluency, and relevance. We obtain moderate agreement for relevance in both directions and moderate agreement for coherence and consistency in hDe-En. The annotation agreement for hEn-De is slightly worse at a weak-moderate level except for relevance. One exception is fluency in hDe-En, where the correlation is poor.

We report the mean value of human and ChatGPT (GPT-4-1106-preview) annotations for each model in Table 11.[15] The best scores for *supervised abstractive models* in both directions range between 2 to mildly above 3 by human annotators and below 2 by ChatGPT, which indicates the quality of outputs is bad to mediocre. However, the highest scores for ChatGPT as a summarizer are over 4 (very good) for hDe-En and 3 to 4 (mediocre to good) for hEn-De by human annotators. Overall, the quality of outputs for hEn-De is lower than that for hDe-En. This is also expected since the length of historical English documents is much longer with 9,643 words on average, which results in more

[15] We utilize both GPT-3.5-turbo and GPT-4-1106-preview for annotation. GPT-4-1106-preview correlates better with human annotators than GPT-3.5-turbo. We report the annotation results from GPT-4-1106-preview in Table 11 in the main text. The annotation results from GPT-3.5-turbo are reported in Table 21 in Section 8.6 (Appendix).

Table 11
Average human and ChatGPT (GPT-4-1106-preview) ratings for the CLCTS dataset (in panels (a) and (b), the scores from human and ChatGPT are separated by a slash (/), i.e., human-annotation/ChatGPT-annotation). Panel (c) is the document level annotation agreement between humans and ChatGPT. The best result of all models is in bold font. The best score for the supervised abstractive model is indicated with an underline.

Model	Coherence	Consistency	Fluency	Relevance
Supervised Extractive				
MemSum				
-translation$_{\text{max 25}}$	2.52/1.60	2.92/1.50	2.98/1.20	3.32/1.80
-Norma-translation$_{\text{max 25}}$	2.50/1.50	2.92/1.40	2.95/1.10	3.20/1.70
Supervised Abstractive - Baseline CLCTS				
mLED				
-Base	2.10/1.62	2.08/1.12	2.78/1.50	2.22/1.38
-Historical MT	2.86/1.67	2.63/1.56	3.01/1.78	3.19/1.67
-Bidirection$_{\text{Prefix}}$	2.43/1.30	2.28/1.10	2.54/1.30	2.58/1.20
Supervised Abstractive - intermediate fine-tuning				
mLED				
-MLS$_{\text{tgt}}$	2.75/1.22	2.56/1.11	3.08/1.22	3.39/1.22
-MLS$_{\text{src+tgt}}$	2.38/1.00	2.30/1.00	2.78/1.10	2.75/1.00
-MLS-CLS	2.20/1.00	1.90/1.00	2.48/1.00	2.42/1.00
-MLS-CLS-CTS	2.11/1.10	1.92/1.00	2.30/1.10	2.28/1.10
Zero-shot Abstractive - ChatGPT				
ChatGPT as summarizer				
-e2e (Title) prompt	2.55/1.70	3.05/1.20	4.10/3.70	2.02/1.40
-e2e prompt	4.14/**3.30**	3.98/**2.40**	4.18/3.80	3.97/2.70
-pipeline prompt	**4.35**/**3.30**	**4.30**/**2.40**	**4.55**/**4.00**	**4.35**/**3.00**
-retrieve-ChatGPT$_{\text{max 100}}$	4.08/2.90	4.08/2.10	4.15/3.50	4.25/2.70

(a) Direction hDe-En

Model	Coherence	Consistency	Fluency	Relevance
Supervised Extractive				
MemSum				
-translation$_{\text{max 25}}$	2.47/1.50	2.34/1.25	2.62/1.38	2.19/1.38
-Norma-translation$_{\text{max 25}}$	2.53/1.38	2.12/1.25	2.75/1.38	2.22/1.25
Supervised Abstractive - Baseline CLCTS				
mLED				
-Base	2.01/1.00	1.56/1.00	2.32/1.00	1.65/1.00
-Historical MT	1.98/1.38	1.78/1.25	2.74/1.38	1.78/1.25
-Bidirection$_{\text{Prefix}}$	2.75/1.38	2.03/1.12	2.78/1.25	1.97/1.25
Supervised Abstractive - intermediate fine-tuning				
mLED				
-MLS$_{\text{tgt}}$	1.99/1.12	1.72/1.12	2.43/1.12	1.91/1.12
-MLS$_{\text{src+tgt}}$	1.83/1.00	1.70/1.00	2.33/1.00	1.66/1.00
-MLS-CLS	1.80/1.00	1.71/1.00	2.39/1.00	1.98/1.00
-MLS-CLS-CTS	2.41/1.00	1.88/1.00	2.34/1.00	2.00/1.00
Zero-shot Abstractive - ChatGPT				
ChatGPT as summarizer				
-e2e (Title) prompt	3.31/2.00	2.25/1.00	3.28/3.50	2.31/1.38
-e2e prompt	3.31/**3.00**	2.80/2.22	3.38/3.56	2.98/2.33
-pipeline prompt	3.34/2.86	**3.19**/2.00	**3.47**/3.57	**3.31**/2.57
-retrieve-ChatGPT$_{\text{max 100}}$	**3.38**/**3.00**	2.94/**2.38**	3.34/**3.75**	3.06/**2.62**

(b) Direction hEn-De

Models	Coh.	Con.	Flu.	Rel.
hDe-En	0.559	0.458	0.755	0.599
hEn-De	0.657	0.650	0.539	0.672

(c) Annotation agreement for human and ChatGPT

information loss due to text truncation because of computational limitations[16] and the cross-lingual embedding distance from Table 3 also suggests more difficulty of hEn-De compared to hDe-En.

For both directions, humans give higher scores to summaries generated by ChatGPT, especially the ChatGPT pipeline prompt for hDe-En with an increase of annotation score to a great margin. Unlike the evaluation metrics, humans prefer pipeline prompt output over e2e prompt output for hDe-En and regarding consistency, fluency, and relevance for hEn-De. For comparing fine-tuned models, humans give higher scores for the model fine-tuned with historical translation in coherence and consistency similar to evaluation metrics for hDe-En. The ratings for hEn-De differ where human ratings prefer mLED-Bidirection$_{\text{Prefix}}$. Since the overall quality of generated summaries is worse for hEn-De, the comparisons among outputs may be more difficult for both humans and evaluation metrics.

Mixed Results for ChatGPT (GPT-4-1106-preview) as an Evaluator. Table 21 (c) shows the annotation agreement between humans and ChatGPT annotation. For both directions, the agreement is moderate to good overall with a slightly better agreement for direction hEn-De (except for fluency). Similarly to the findings from Chiang and Lee (2023), we observe lower Likert scores from ChatGPT evaluation compared with human evaluation except for summaries generated by ChatGPT itself. However, the output for hEn-De is more concerning where we observe more ratings of score 1. Another point worth noticing is that even though in our instruction, we allow for 0.5 increments, we do not observe such ratings from ChatGPT output.

To sum up, ChatGPT as an evaluator (1) is prone to lower scores than humans;[17] (2) cannot distinguish summaries of low quality (hEn-De); and (3) does not fully follow the instruction of 0.5 score increments.

5.3 Example Outputs

In this section, we provide four sample outputs (for direction hDe-En) from both abstractive e2e models and ChatGPT, together with the corresponding reference texts (see Section 8.7 for a more detailed analysis). We use two documents as examples; and for each document, we list one generated summary of mediocre to good level quality and one of low quality according to human annotation. The results are shown in Table 12. Based on the error taxonomy from Goyal, Li, and Durrett (2022b), we highlight factual inconsistency and language errors (unclear coreference of pronouns, nonsensical text and untranslated expression) by different colors.

For low quality generation (summary 1.2 and summary 2.2), we observe a substantial amount of factual inconsistency with the source text (1) and nonsensical text (3). In summary 2.2, we also notice the untranslated expression “Nimmersatt” repeatedly appears in the text. Unclear coreference of pronouns is another issue that affects the readability of the outputs. For example, in the sentence from summary 1.2 “the priest asks him if he (2) can come to the wedding”. The pronoun “he” is ambiguous as it could refer to the boy or the priest himself. In contrast, summary 1.1 and summary 2.1

[16] We also experimented with mBART, fine-tuning with Unlimiformer under both the low-cost training and computationally expensive long-range training method. Since the low-cost training method only sees full texts at test time and long-range training suffers from the same GPU memory constraints as mLED, we do not observe improvements in model performance compared to baseline mLED.

[17] We include one sample annotation from the gold standard annotations during instruction.

Table 12
Example of generated summaries and the reference texts. Sentences highlighted in orange contain factual inconsistency with the source text (1). Unclear coreference of pronouns (2), nonsensical text (3), and untranslated expression (4) are highlighted in blue. Spelling, grammar, and punctuation mistakes are highlighted in red and bold.

Reference Texts
"A poor peasant boy hears a sermon that the way to heaven always leads straight ahead. He goes and comes to the service in a big city church, thinks he is in heaven and cannot be sent away. He sees people praying in front of Mary with the baby Jesus and thinks it is God. He shares his food with him, and the image grows thicker. Once, when he is bedridden for eight days, he goes and apologizes to the image. It invites him to the wedding next Sunday. The priest, who has been watching him, is not allowed to go, but gives him communion on Sunday, when the boy falls down dead."

1.1 Generated Text from model mLED-Base, quality level 3-4
"A poor peasant boy wants to go to Heaven, but the parson tells him to go straight. When he arrives in the church, he thinks God is there, and says, "Listen, dear God, I will bring you half of my food every day." Every day he cooks (1) half of his food, and the image begins to eat. A couple of weeks later, when people realize that the image is getting bigger and bigger, he stays in church and sees his bread being shared with the mother God. He gets sick and can't get up for eight days. The parson goes after him and hears him say, "God, don't take it so long (3), I was sick and **could't** stand up." He answers (2), "That doesn't matter**,** I've seen your good will**,** that's enough, and next Sunday you are to come to my wedding." The boy rejoices and asks for the image (1). "No, it's you alone," says he. He (2) first wants to prepare and gives him the supper. The next Sunday, when supper is coming to him, he falls down and is dead and is **to eternal wedding**." "

1.2 Generated Text from model mLED-MLS-CLS-CTS , quality level 1-2
"A poor peasant in the church hears God (1) admonishing him to go to Heaven, and he goes to God's church. The priest tells him not to go out, but he wants to keep him (1). He stays in **church** and when he sees the mother God statue, which is cut from wood, the little Jesus (3), he says that it's God, which means that he will eat half of his food every day (1). After a few weeks, people marvel at the statue and see how he spreads bread with God (1). After eight days, the priest asks him if he (2) can come to the wedding. The boy replies, "No, you're only me, I told you that you told me that you were you alone. I'll give him dinner first, which he says he will give him, and then he dies, then he'll never be able to marry again," so he has to go back to the church and asks if he could come with God alone. After the wedding, the boy asks the parson to give him the dinner, but God says he's dead and he is dead, and God says that he must marry him again (3)."

Example 1

Reference Texts
"The wolf is the stronger of the two and dominates the fox. The fox would like to be rid of him. The wolf forces the fox to get him food. He gets him a lamb and leaves. But the wolf is even hungrier and tries to get a lamb for himself. But he is so clumsy that the farmers notice and beat him. The wolf says to the fox that he has tricked him. The fox replied, 'Why are you such a glutton!' The next day the fox has to get pancakes for the wolf. They go to the house together and the fox sneaks around the house to carefully get six pancakes from the plate. The wolf again doesn't have enough and wants to get more. But he again doesn't manage and the farmer's wife beats him again. The following day, the two go to the butcher. They jump through a window hatch into the cellar. The wolf eats a lot. The fox, on the other hand, eats little and keeps looking to see if he still fits through the window hatch. Suddenly the butcher comes, the red fox jumps out quickly, only the wolf gets stuck and is beaten to death by the butcher."

2.1 Generated Text from model ChatGPT-e2e prompt, quality level 4-5
"The wolf and the fox were walking in the woods together. The wolf demanded that the fox bring him food or he would eat him. The fox suggested they steal a lamb from a nearby farm. The wolf ate the lamb and wanted more, but when he tried to steal another, he was caught and beaten by the farmers. The next day, the fox suggested they steal pancakes from a farm, but the wolf ate too many and was caught again. On the third day, the fox suggested they steal salted meat from a cellar, but the wolf ate too much and got stuck in the cellar. The farmers (1) found him and killed him, while the fox escaped and was happy to be rid of the greedy wolf."

2.2 Generated Text from model mLED-MLS$_{src+tgt}$, quality level 1-2
"The wolf wants to catch the fox because he is the weakest and would be the master. (1) The fox tells the wolf to get a lamb. (1) The lamb is not happy, (3) but the other wants to have it too. The farmers run to get the lamb, and they hit him (2). When they come to rescue them, the witch tells them they are a nimmersatt (3, 4). They go to a farmer's house, where they (1) cook pancakes, and the sheep eat the pancakes again. (1) The sheep shout loudly and **calls** for help, but when she comes out, she says that they have been slaughtered by a man in the forest, and he says that he wants to help out. (3) The **wolfe lets go** and jumps through the hole, but he does not want to see anything. He jumps out of the hole and kills the shepherd, who does not know what to do. (3) On the third day, he (2) is happy that he is rid of the old Nimmersatt (4), who has been rid of a man who is slaughtering meat in the cellar, and who is the first to get rid of him. (3)"

Example 2

contain fewer errors, especially the ChatGPT output which only contains one factual inconsistency. In summary 1.1, we find two factual inconsistencies with the source text where the boy "shares" the food instead of "cooking" it himself and he "asks" the image instead of "asking for" it. Additionally, we also notice several other errors such as unclear coreferences of pronouns, nonsensical texts, and spelling mistakes (such as "could't").

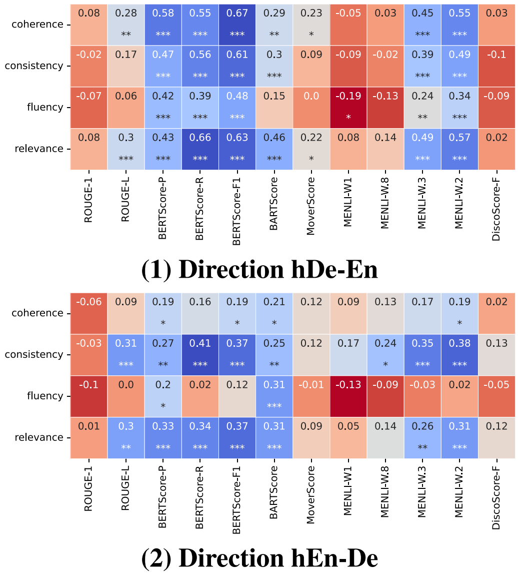

Figure 5
Segment level Spearman's ranking correlation between human annotation and evaluation metrics. The corresponding p-value is obtained through a two-tailed Student t-test at significance levels 0.05 (*), 0.01 (**), and 0.001(***).

6. Discussion and Analysis

6.1 How Well Do Different Metrics Perform at the Segment Level?

Figure 5 shows Spearman's ranking correlation between human annotation and evaluation metrics at the segment level. Recall that the correlation results between GPT-4 and human annotation are provided in Table 11 (see Table 21 for GPT-3.5).

Decent Correlation from BERTScore Variants but Worse for German. For hDe-En, we observe a weak to moderate level of correlation with human annotation for BERTScore, BARTScore, MENLI-W$_{.3}$, and MENLI-W$_{.2}$. MoverScore correlates weakly in coherence and relevance and the same is true for ROUGE-L. ROUGE-1 has low correlations with human annotation and the same applies for MENLI-W$_{.8}$ and DiscoScore. The situation is slightly worse for hEn-De, where we observe a lower level of correlation in general compared to hDe-En and less significant correlations from BARTScore and BERTScore variants according to the Student t-test, especially for coherence. ROUGE-L correlates moderately for direction hEn-De in consistency and relevance. Overall, BERTScore variants correlate slightly better than other metrics with humans in hDe-En for coherence (0.67), consistency (0.61), and relevance (0.66), and in hEn-De for consistency (0.41) and relevance (0.37).

6.2 Statistical Analysis on Model Performance

To quantify the effect of document level characteristics on model performance and contribute to explainability of our models, we analyze how BERTScore-F1 as our best automatic metric overall changes depending on four document features reported in

Table 13
Regression results for document features (where β_S, β_L, β_D, and $\beta_{Year:*}$ represent coefficients for variable Similarity, Length, MDD, and Year respectively). The corresponding p-value is obtained through a two-tailed Student's t-test for coefficients at significance levels 0.05 (*), 0.01 (**), and 0.001(***). "Base" indicates the base group which is represented by the intercept. For hEn-De, the base group for publication year is Year grouping before 1800 (Year:$-$1800).

Direction	Numerical			Categorical	
	β_S	β_L	β_D	$\beta_{Year:1800-1850}$	$\beta_{Year:1850+}$
hDe-En	0.06^{***}	-0.14^{***}	0.03^{*}	base	-0.05
hEn-De	0.24^{***}	-0.15^{***}	-0.08^{***}	-0.003	0.69^{***}

Section 3, namely, the mean length of documents (Length), mean dependency distance (MDD), publication year (Year), and divergence of sentence semantics between document and summary by computing the mean sentence-level cosine similarity (Similarity). We apply normalization to numerical features including BERTScore-F1, Length, MDD, and Similarity to have zero mean and unit variance.[18] We use publication year as a categorical variable where we categorize it by two time periods for hDe-En, namely, 1800–1850 (Year:1800–1850) and after 1850 (Year:1850+), and for hEn-De we add another grouping which includes all documents before 1800 (Year:$-$1800). To examine model effects, we also encode a categorical variable to represent different models (Model). The regression formula is thus:

$$\text{BERTScore-F1}_i = \beta_0 + \beta_S \text{Similarity}_i + \beta_L \text{Length}_i + \beta_D \text{MDD}_i + \beta_{Y_j} \text{Year}_i + \beta_{M_k} \text{Model}_i + \varepsilon_i \quad (1)$$

where i represents documents, j represents year groups, and k represents models defined in Section 4 and $\varepsilon_i \in \mathcal{N}(0, \sigma)$ is an error term. We fit Equation (1) using ordinary least squares via statsmodels (Seabold and Perktold 2010) separately for both directions (hDe-En and hEn-De).

Effect of Document Features. We report the regression results of document features in Table 13.[19] For both directions, **embedding similarity** measured by β_S has a significantly positive effect on model performance, which indicates the model performs better with higher embedding similarity between document and summary. In contrast, **the mean length of document** (measured by β_L) shows a significantly negative relation to model performance where more tokens in the source document result in worse performance. This is expected since longer inputs increase the difficulty of summarization and also increase the chance of information loss due to input truncation. The effect of **mean dependency distance** (β_D) varies. We obtain a significantly negative effect for hEn-De, which indicates that sentences with more syntactic complexity have a negative influence on model performance. A positive effect for hDe-En is obtained but with a low level of significance. The effect of **document publication year** is not significant for hDe-En where the model performance differs insignificantly between base group Year:1800–1850 and group Year:1850+. For hEn-De, this also holds for documents published in 1800–1850 compared with the base group with the publication year before 1800.

[18] We exclude the concern of multicollinearity for numerical variables by computing variance inflation factors and the details are reported in Section 8.8 (Appendix).

[19] The adjusted R^2 is 0.425 for direction hDe-En and 0.551 for direction hEn-De.

However, we observe a significantly positive effect on model performance if we compare the group Year: 1850+ to base group Year:−1800. This effect also coincides with our finding from Section 3 that English documents from the late 1800s share more similarities to modern English, and thus have a positive impact on model performance.

Also worth noticing is that the main contributor among document features to model performance varies between directions. For hDe-En, the main influence comes from the length of document and mildly from embedding similarity while for hEn-De, and the main contributor is the publication year (group Year:1850+), where embedding similarity and the length of document rank second and third among document features, respectively. Overall, however, our findings are largely consistent and they are meaningful: As we argued in the introduction, older texts are more difficult to handle, and the same is true for longer and more syntactically complex source documents; more generally, the more divergence between source documents and reference, the harder is the summarization task.

Effect of Model Choice. The effect of model choice coincides with our findings from Section 5 where, for both directions, we obtain a significant gain with ChatGPT models (except for the ChatGPT title prompt) and, in contrast, extractive models (extract-then-translate) are significantly worse than the baseline model **mLED-Base**. For hDe-En, the ChatGPT title prompt model performs significantly worse than the baseline **mLED-Base**, though not significant for hEn-De. We also notice that for hDe-En, models with intermediate fine-tuning provide significantly worse performance compared to the baseline as discussed in Section 5; and for hEn-De, the improvement of model performance adding intermediate fine-tuning tasks is not significant. We include the complete results in Section 8.8 (Appendix).

Effect of Training Set Size. As discussed in Section 3 and Table 2, the size of our corpora is much smaller compared with other datasets such as CNN/Daily Mail, Wikilingua, and MLSUM. To investigate whether the limited amount of data for fine-tuning is the main cause of bad performance, we harvest additional instances for both directions. Specifically, we expand the existing datasets for hDe-En from 328 to 455 instances (38.6% increase) and for hEn-De from 289 to 501 instances (73.4% increase). The newly added sources for hDe-En mainly include 95 chapters from the philosophical novel *Wilhelm Meister's Apprenticeship* published in 1795, 12 books of epic poetry *Reineke Fuchs* published in 1794, and 15 scenes from Friedrich Schiller's play *The robbers* published in 1781. For hEn-De, the new sources consist of 210 historical translations of fairy tales from different parts of the world published between 1885 and 1910 and 2 short stories.[20] We retrain the CLCTS baseline models using the expanded datasets for both directions, using the same methods described in Section 4. To ensure comparability, we perform automatic evaluations on the same test sets. Table 19 shows the results for both directions. For hDe-En, the baseline models exhibit marginal improvements, with a mean increase of 0.004 in ROUGE-1, 0.010 in ROUGE-L, and 0.008 in BERTScore-F1. Conversely for hEn-De, the model performances stay invariant on average according to ROUFE-L and BERTScore-F1 with a slight decrease according to ROUGE-1 (−0.004). We speculate that the decrease is because the original CLCTS hEn-De datasets contain more diverse sources (fairy tales, short stories, and plays) while the newly acquired

[20] The publication time of the newly acquired datasets for both directions closely aligns with that of the original dataset discussed in Section 3.

Table 14
Counts of invalid outputs from ChatGPT. Tgt.lang is the target language. Invalid/Obs. represents invalid cases/ total observation.

Model	Tgt.lang	Invalid/Obs.
ChatGPT-title	De	1/328
ChatGPT-pipeline	De	34/328
ChatGPT-e2e	De	57/328
ChatGPT-title	En	0/289
ChatGPT-pipeline	En	2/289
ChatGPT-e2e	En	8/289

training set primarily contains fairy tales. Moreover, as demonstrated in Section 6.1, the correlation between automatic evaluation metrics and human annotation is worse for hEn-De, which may also affect the evaluation. Overall, our experiments with a larger training set size only show marginal improvements to the existing baseline models. Based on this observation, we hypothesize that a greater volume of data might be necessary to effectively address the intricate nature of the CLCTS task.

6.3 Analysis on ChatGPT Summarization

6.3.1 The Effect of Different Prompts and Target Language. A case study with ChatGPT-e2e vs. ChatGPT-pipeline. Though in Section 5 we observe better metric and human annotation results from the ChatGPT e2e prompt, we notice that ChatGPT under this prompt is prone to output summaries in the wrong target language (invalid output). In such cases, we query the summary with the same prompt until a valid output is returned. We calculate the occurrences of *invalid* outputs after 2 rounds of queries and report them in Table 14, where we observe that using e2e prompts has a higher chance to yield invalid outputs. What is also worth noticing is that, when the output is expected in German, we observe more invalid cases among all three prompts; this means that ChatGPT tends to output English texts when our expected language is in German.

6.3.2 ChatGPT for Historical Text Translation. We conduct an additional analysis and observe a performance gain of 5% average over all metrics by simply switching the translator to ChatGPT. Part of the reason is that ChatGPT can translate better with prior knowledge, for example, ChatGPT keeps the word “Schimmel” (a special type of horse) unlike other translators translating to the meaning “mold.” What is also worth noticing is that ChatGPT can better process historical variants such as historical spellings (e.g., aͤ, oͤ, uͤ) and expressions, which is very suitable for our setup. This could fill the gap of historical spelling normalization tools as discussed in Section 5.

6.3.3 ChatGPT Knows It All? Recently, the prevalence of LLMs has raised the concern of data contamination (Balloccu et al. 2024; Sainz et al. 2023). Because our corpus-building process is highly dependent on Wikipedia (and popular stories presumably discussed on other places on the Web), we want to know to what extent ChatGPT has an unfair advantage. We ask ChatGPT to output summaries based on the author, title, and publication year. ChatGPT reports summaries to all requests in both language directions, though the output may contain irrelevant information besides the actual summarization.[21]

However, despite the overall outstanding performance of ChatGPT summarization according to humans, ChatGPT evaluation, and automatic evaluation metrics, ChatGPT as a summarizer can also be prone to hallucinations like other NLG systems (Bang et al. 2023; Ji et al. 2023). For example, in the summary of the famous story *Cinderella*, we notice the golden slippers in the original document become glass slippers, which is the best-known version worldwide but is incorrect based on the input text.

Adversarial Attacks for ChatGPT. Knowing that ChatGPT has specific prior knowledge of our CLCTS task, we go a step further and design three adversarial experiments inspired by the work from Chen and Eger (2023) where we gradually reduce the amount of specific prior knowledge in the input documents (from seen documents to pseudo-unseen documents to unseen documents). We aim to answer (1) does ChatGPT summarize truly based on the input text given its specific prior knowledge? (2) Can ChatGPT summarize against its specific prior knowledge? (3) How well can ChatGPT summarize facing unseen documents?

For question (1), we design a *sentence omission task*. By dropping an increasing amount of sentences from the original documents, we control the information reduction of ChatGPT input for summarization. We randomly select 11 historical English documents from our CLCTS corpus containing 100-150 sentences and construct new documents by dropping sentences of varying percentages. We then use ChatGPT to summarize the new documents. To quantify the information change, we use the summary of the original documents (generated by ChatGPT) as the reference and evaluate generated summaries from documents after omission using automatic evaluation metrics.[22] We average the resulting metric scores after scaling and use it as our similarity measure. The result is shown in Figure 6. We observe an overall downward trend of similarity score of both CTS and CLCTS, which reflects a certain, level of information loss accordingly.

For question (2), we first construct pseudo-unseen documents by entity swap, negation, and omission.[23] *Entity swap:* We swap multiple entities from the original story with new entities. For example, in the story of Cinderella, we switch the hazel tree into a cactus, and (to make it funny) we replace the prince with a watermelon (to ensure coherence, we also add background such as "They live in the Fruit Kingdom."). *Negation:* We alter the important aspects of the plot to make a "negated story." For example, in the story of Cinderella, the stepmother and stepsisters are mistreated by the rich man's evil daughter and the prince eventually marries one of the stepsisters. *Omission*: For omission, we omit important plots. For example, Cinderella neither attends the ball nor marries the prince. We make sure each pseudo-unseen document only contains one kind of attack. We manually annotate the resulting output where we focus on whether the attack is completed accurately or not (e.g., if during negation, Cinderella is still mistreated by the stepmother—as in the original story—then the task is judged as a

[21] For example, "The story emphasizes the power of kindness and the importance of inner beauty. It also highlights the value of perseverance and the rewards that can come from staying true to oneself."

[22] We set the temperature to 0 since our goal is not to obtain the best summaries but to test ChatGPT summarization performance under gradual information loss.

[23] For each attack, we construct 6 documents. We list examples of adversarial attacks in Section 8.9. For each document, we query the summaries 3 times at temperatures 0, 0.7, and 1 for both English and German. After excluding summaries in the wrong target language, we have 300 valid generated summaries.

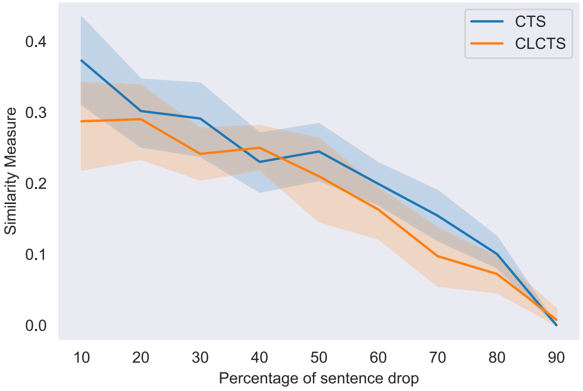

Figure 6
Similarity measure of generated summaries compared to the original summary over different percentages of sentence drop. We show an average score over all documents and present 95% confidence interval.

Table 15
Accuracy of different attacks for CTS and CLCTS tasks for ChatGPT.

Attack	CTS	CLCTS
Omission	0.79	0.67
Entity Swap	0.83	0.71
Negation	0.76	0.53

failure).[24] We compute the accuracy and report it in Table 15. We observe the following from the results. (1) ChatGPT can better handle omission and entity swap than negation. This may be because, compared with omission and entity swap, negation attacks alter the underlying logical connections of the story and thus require more reasoning to complete this attack under the assumption that ChatGPT is affected by its explicit knowledge during summarization. (2) We find higher accuracy for CTS than CLCTS task. For all three attacks, we observe a higher accuracy of completing the task for CTS. This is understandable since CLCTS also involves cross-lingual summarization, which is more difficult.

To further investigate question (3), we create 15 unseen documents from fiction chapters whose last updates are in the year 2022 or later.[25] For modern fiction chapters, we translate and rewrite the stories in historical languages (by adding historical language elements such as morphological changes) and write reference summaries in modern languages. Both documents and summaries are proofread by a co-author of this work. Similar to Section 5, we evaluate the resulting summaries by three human annotators from four perspectives. The results are shown in Table 16. We notice that the overall rating for CTS and CLCTS ranges between 3 and 4, indicating that ChatGPT summarizes unseen documents at a mediocre level—which is similar to the results from

[24] We obtain the inter-annotator agreement between two annotators of 0.7 according to Cohen's kappa and 0.87 according to percentage accuracy based on 15 samples. We list examples of failure cases in Section 8.10 (Appendix).

[25] We include chapters from five Chinese novels and ten English novels.

Table 16
Average human ratings for ChatGPT summarization on unseen documents with respect to coherence, consistency, fluency, and relevance.

Models	Coh.	Con.	Flu.	Rel.
unseen CLCTS hEn-De	3.24	2.79	3.06	2.85
(our CLCTS hEn-De	3.31	2.80	3.38	2.98)
unseen CTS hEn-En	3.44	3.12	3.44	3.29
(our CLCTS hDe-En	4.14	3.98	4.18	3.97)

seen documents. This rating over all four dimensions for hEn-De is slightly worse than the results from our previous experiment using CLCTS corpus (we list the result of the same direction in row 3 in Table 16). We also query and annotate English summaries of unseen documents (CTS hEn-En), where we observe the overall performance is better than CLCTS hEn-De for unseen documents. This is expected since the CTS task does not involve cross-lingual summarization. However, we observe a decrease in summary quality of 0.75 points on average compared with the annotation results from the CLCTS hDe-En task from our experiment. This is slightly surprising, since both target languages are in English, and especially CLCTS hDe-En has a more complex task setting. This may again indicate the benefits of the specific prior knowledge from ChatGPT. Overall, we argue that ChatGPT can clearly outperform fine-tuned models for CLCTS tasks in our experiments. However, it does not "magically" solve all the issues, as per our experiments: (1) it profits from its prior knowledge to a certain degree where it may also blend its prior knowledge with the observed data and ChatGPT is better at omission and entity swap than negating against its prior knowledge; (2) the benefit from prior knowledge slightly inflates its assessed quality where ChatGPT generates summaries of mediocre quality from unseen source documents with a slightly worse performance over all four evaluating dimensions compared to seen documents for hEn-De; and (3) the performance is language dependent where, compared with German outputs (hEn-De), English outputs (hEn-En) yield slightly better ratings.

7. Concluding Remarks

In this work, we build the first CLCTS corpus and a historical translation dataset with additional information for future exploration. We study the characteristics of this corpus both alone and in comparison to other summarization corpora and examine the effectiveness of pipeline models, popular e2e transformer-based abstractive models leveraging intermediate task fine-tuning, and ChatGPT for the CLCTS task. Even though these methods have been shown effective for single-task settings such as CLS or long document summarization, *they fail to deliver good summaries in our combined task setting (cross-lingual, cross-temporal, and long document)*. We showcase the potential of GPT-3.5 in CLCTS as a zero-shot summarizer. It provides moderate to good quality outputs and seems very adept as a context-aware spelling normalization tool. However, it profits from its prior knowledge to a certain degree as tested by adversarially attacked and unseen source documents. Overall, we observe better performance for plot omission and entity swap than plot negation against its prior knowledge. Moreover, GPT-3.5 performs slightly worse for unseen source documents compared with seen documents.

As for evaluation, ChatGPT (both GPT-3.5 and GPT-4) as an evaluator can correlate with human evaluations at a moderate to good level but is prone to giving lower scores.

Additionally, we find *a moderate correlation of BERTScore with human annotation (slightly worse for German)*.

By regression analysis with BERTScore-F1, we quantify the effects of document features on model performance, where we find positive impacts from documents with more recent publication years and document-summary embedding similarity, while we observe a negative relation between model performance and the length of document. This helps explain model performance and affirms our hypotheses about the difficulty of the CLCTS task as discussed in the Introduction (language change and divergence, etc.).

Future work can address the limitations revealed by our experiments. For example, researchers can explore other recent models of great potential, such as Unlimiformer under sufficient GPU memory (Bertsch et al. 2024) and Longnet (Ding et al. 2023). Other training frameworks such as pretraining under multi-task learning are also promising alternatives. Incorporating tasks such as non-parallel historical corpora for pretraining and the PAR3 cross-temporal dataset for machine translation (Thai et al. 2022) may further provide benefits. Finally, future work can focus on extending the size and diversity of our CLCTS datasets, including more diverse language pairs.[26]

8. Appendix

8.1 Links to the Sources

- DTA: `https://www.deutschestextarchiv.de/`
- Wikisource: `https://www.wikisource.org/`
- Examples of proofreading:
  `https://de.wikisource.org/w/index.php?title=Seite:Kinder_und_Hausm%C3%A4rchen_(Grimm)_1812_I_008.jpg`
- Project Gutenberg: `https://www.gutenberg.org/`
- Beautifulsoup library: `https://pypi.org/project/beautifulsoup4/`.
- list of German fairy tale: `https://de.wikipedia.org/wiki/Liste_von_MÃČÄŔchen#Deutsche_MÃČÄŔchen`
- list of English short stories:
  `https://en.wikipedia.org/wiki/Category:Short_stories`
- WikiLingua Repository: `https://github.com/esdurmus/Wikilingua`
- CNN/Daily Mail Dataset:
  `https://huggingface.co/datasets/cnn_dailymail`

[26] Our current work builds the CLCTS corpus for the high-resource language pair German and English. Main difficulties working with other languages include: (1) Even more Wikipedia summaries may be missing for other languages (German and English are two of the highest resource languages); (2) it may be more difficult to obtain historical sources of other languages; (3) human evaluation of other languages may be less accessible and thus even more costly; and (4) automatic metrics including GPT4 may also be worse as we have observed worse performance for German than for English. While exploring more distant language pairs is thus immensely interesting, (5) our results further indicate that transfer would be even more difficult for all models involved.

8.2 Example of Translation dataset

Table 17 showcases an example of the translation dataset.

8.3 Configuration of Evaluation Metrics

Table 18 shows the configuration of evaluation metrics.

8.4 Results for Baseline CLCTS Trained with the Expanded Datasets

Table 19 shows the results for baseline CLCTS hEn-De and hDe-En using the expanded datasets.

8.5 Annotation Details

We utilize the same evaluation criteria as SummEval (Fabbri et al. 2021). Annotators are required to read the instructions listed below carefully. Then, they are provided with a source document, a corresponding reference summary, and a generated summary for evaluation. All annotators are blind to the model information (i.e., annotators do not

Table 17
Example of translation dataset. We highlight words with spelling/morphological changes in *italics*.

German Text: Läuschen und Flöhchen
”Ein Läuschen und ein Flöhchen die lebten zusammen in einem *Haushalte* und brauten das Bier in einer Eierschale. Da fiel das Läuschen hinein und verbrannte sich. Darüber *fieng* das Flöhchen an laut zu schreien. Da sprach die kleine *Stubenthüre* ‘was schreist du, Flöhchen?’ ‘Weil Läuschen sich verbrannt hat.’ Da *fieng* das *Thürchen* an zu knarren. Da sprach ein Besenchen in der Ecke ‘was knarrst du, *Thürchen*?’ ‘Soll ich nicht knarren?”’ [...]
English Text: The Louse and the Flea
”A louse and a flea kept house together and were brewing beer in an egg-shell. Then the little louse fell in and burnt herself. On this the little flea began to scream loudly. Then said the little room-door, ”Little flea, why art *thou* screaming?“” Because the louse has burnt herself.” Then the little door began to creak. On this a little broom in the corner said, ”Why art *thou* creaking, little door?“”*Have I not* reason to creak?””[...]

Table 18
Configuration of evaluation metrics.

Metrics	Eval Lang	Configurations
BERTScore	English	microsoft/deberta-xlarge-mnli
	German	xlm-roberta-large
MoverScore	English	distilbert-base-uncased
	German	xlm-roberta-large
BARTScore	English	facebook/bart-large-cnn
	German	facebook/mbart-large-50-many-to-many-mmt
MENLI	English	NLI-D default, microsoft/deberta-large-mnli
	German	NLI-D cross-lingual, MoritzLaurer/mDeBERTa-v3-base-mnli-xnli
DiscoScore	English	Conpono
	German	bert-base-multilingual-cased

Table 19
Results for Baseline CLCTS hEn-De and hDe-En using the expanded datasets (hDe-En from 328 to 455 instances and hEn-De from 289 to 501 instances). The corresponding changes compared to the results in the main text (i.e., new results minus the old results) are reported in parentheses.

Model	**ROUGE-1**	**ROUGE-L**	**BERTScore-F1**
	Supervised Abstractive - Baseline CLCTS		
mLED			
-Base	0.389 (−0.002)	0.206 (0.004)	0.552 (0.005)
-Historical MT	0.397 (0.003)	0.213 (0.013)	0.561 (0.009)
-Bidirection$_{\text{Prefix}}$	0.398 (0.010)	0.211 (0.013)	0.558 (0.011)
Avg. change	0.004	0.010	0.008

(a) Direction hDe-En

Model	**ROUGE-1**	**ROUGE-L**	**BERTScore-F1**
	Supervised Abstractive - Baseline CLCTS		
mLED			
-Base	0.315 (−0.007)	0.141 (−0.001)	0.847 (−0.001)
-Historical MT	0.329 (0.001)	0.145 (0.002)	0.849 (0.001)
-Bidirection$_{\text{Prefix}}$	0.323 (−0.005)	0.144 (−0.001)	0.848 (−0.001)
Avg. change	−0.004	0.000	0.000

(b) Direction hEn-De

know which output comes from which model). The three authors of this paper (out of six annotators) participated in the annotation as well and they have as much information as the other three annotators. None of the annotators have a priori preferences for any of the models examined. The selection of model outputs for each source document is random and we shuffle the documents before they are presented to the annotators. We ask the annotators to rate the summaries on a Likert scale from 1 to 5 (higher better, including 0.5 increments) along the four dimensions, namely, coherence, consistency, fluency, and relevance.

The following guidelines are presented to the annotators before annotation:

Information and Instructions

In this task, you will evaluate automatically generated summaries of historical short stories. The source document and a reference summary from Wikipedia are provided.

- You should evaluate the texts based on their coherence, consistency, fluency, and relevance.
- Rate each item with a score from 1 (worst) to 5 (best).
- In case of uncertainty, the grades can be increased by 0.5 steps (e.g., 3.5).
- A gold standard was defined based on the test phase of the evaluation. Please consider the gold standard; see the corresponding tab.

Definitions (SummEval)

- **Coherence**: The rating measures the quality of all sentences collectively. The sentences must fit together and sound natural. Consider the quality of the summary as a whole.
- **Consistency**: The rating measures whether the facts in the summary are consistent with the facts in the reference summary. Consider whether the summary reproduces all facts accurately and does not make up untrue information.
- **Fluency**: This rating measures the quality of individual sentences, whether they are well-written and grammatically correct. Consider the quality of individual sentences.
- **Relevance**: The rating measures how well the summary captures the key points of the article. Consider whether all and only the important aspects are contained in the summary.

Number of Annotated Instances by Models. Table 20 demonstrates the number of annotated instances for each model. Each instance is annotated by one or several annotators.

8.6 ChatGPT Evaluation: Annotation Results from GPT-3.5-turbo

Table 21 reports the annotation results from GPT-3.5-turbo.

Table 20
Number of annotated instances per model. Each instance is annotated by one or several annotators.

Model	**hDe-En**	**hEn-De**
	Supervised Extractive	
MemSum		
-translation$_{\text{max 25}}$	10	8
-Norma-translation$_{\text{max 25}}$	10	8
	Supervised Abstractive - Baseline CLCTS	
mLED		
-Base	10	8
-Historical MT	10	8
-Bidirection$_{\text{Prefix}}$	10	8
	Supervised Abstractive - intermediate fine-tuning	
mLED		
-MLS$_{\text{tgt}}$	10	8
-MLS$_{\text{src+tgt}}$	10	8
-MLS-CLS	10	8
-MLS-CLS-CTS	10	8
	Zero-shot Abstractive - ChatGPT	
ChatGPT as summarizer		
-e2e (Title) prompt	10	8
-e2e prompt	10	8
-pipeline prompt	10	8
-retrieve-ChatGPT$_{\text{max 100}}$	10	8
Total	130	104

Table 21
Average human and ChatGPT (GPT-3.5-turbo) ratings for the CLCTS dataset (in panels (a) and (b), the scores from human and ChatGPT are separated by a slash (/), i.e., human-annotation/ChatGPT-annotation). Panel (c) is the document level annotation agreement between humans and ChatGPT. The best result of all models is in bold font. The best score for the supervised abstractive model is indicated with an underline.

Model	**Coherence**	**Consistency**	**Fluency**	**Relevance**
Supervised Extractive				
MemSum				
-translation$_{\text{max 25}}$	2.52/2.00	2.92/2.10	2.98/1.80	3.32/1.60
-Norma-translation$_{\text{max 25}}$	2.50/1.90	2.92/1.60	2.95/1.90	3.20/1.20
Supervised Abstractive - Baseline CLCTS				
mLED				
-Base	2.10/1.88	2.08/1.88	2.78/2.00	2.22/1.62
-Historical MT	2.86/1.89	2.63/2.33	3.01/2.00	3.19/2.00
-Bidirection$_{\text{Prefix}}$	2.43/1.70	2.28/1.40	2.54/1.90	2.58/1.20
Supervised Abstractive - intermediate fine-tuning				
mLED				
-MLS$_{\text{tgt}}$	2.75/2.00	2.56/1.89	3.08/2.22	3.39/1.67
-MLS$_{\text{src+tgt}}$	2.38/1.50	2.30/1.30	2.78/1.50	2.75/1.10
-MLS-CLS	2.20/1.40	1.90/1.30	2.48/1.40	2.42/1.10
-MLS-CLS-CTS	2.11/1.60	1.92/1.30	2.30/1.60	2.28/1.20
Zero-shot Abstractive - ChatGPT				
ChatGPT as summarizer				
-e2e (Title) prompt	2.55/3.00	3.05/2.20	4.10/3.70	2.02/2.00
-e2e prompt	4.14/**4.00**	3.98/**4.60**	4.18/**4.10**	3.97/**4.50**
-pipeline prompt	**4.35/4.00**	**4.30**/4.30	**4.55**/4.00	**4.35**/4.20
-retrieve-ChatGPT$_{\text{max 100}}$	4.08/3.60	4.08/3.60	4.15/3.70	4.25/3.90

(a) Direction hDe-En

Model	**Coherence**	**Consistency**	**Fluency**	**Relevance**
Supervised Extractive				
MemSum				
-translation$_{\text{max 25}}$	2.47/1.50	2.34/1.38	2.62/1.50	2.19/1.00
-Norma-translation$_{\text{max 25}}$	2.53/1.62	2.12/1.25	2.75/1.62	2.22/1.00
Supervised Abstractive - Baseline CLCTS				
mLED				
-Base	2.01/1.25	1.56/1.00	2.32/1.50	1.65/1.00
-Historical MT	1.98/2.00	1.78/1.69	2.74/1.94	1.78/1.38
-Bidirection$_{\text{Prefix}}$	2.75/1.69	2.03/1.25	2.78/1.62	1.97/1.12
Supervised Abstractive - intermediate fine-tuning				
mLED				
-MLS$_{\text{tgt}}$	1.99/1.50	1.72/1.38	2.43/1.50	1.91/1.12
-MLS$_{\text{src+tgt}}$	1.83/1.38	1.70/1.06	2.33/1.38	1.66/1.00
-MLS-CLS	1.80/1.29	1.71/1.29	2.39/1.57	1.98/1.00
-MLS-CLS-CTS	2.41/1.12	1.88/1.00	2.34/1.12	2.00/1.00
Zero-shot Abstractive - ChatGPT				
ChatGPT as summarizer				
-e2e (Title) prompt	3.31/2.19	2.25/2.25	3.28/3.19	2.31/1.88
-e2e prompt	3.31/2.78	2.80/2.61	3.38/3.11	2.98/2.33
-pipeline prompt	3.34/**3.25**	**3.19**/3.12	**3.47/3.50**	**3.31/3.12**
-retrieve-ChatGPT$_{\text{max 100}}$	**3.38/3.25**	2.94/**3.38**	3.34/3.25	3.06/3.00

(b) Direction hEn-De

Models	**Coh.**	**Con.**	**Flu.**	**Rel.**
hDe-En	0.512	0.531	0.569	0.549
hEn-De	0.417	0.454	0.424	0.568

(c) Annotation agreement for human and ChatGPT

Table 22 reports the annotation agreement between GPT-3.5-turbo and GPT-4-1106-preview.

8.7 Comparison of the Generated Summaries and References

In Section 5.3, we present references and generated samples showcasing the main errors of the generated outputs. Here, we provide more details comparing generated outputs

Table 22
Document level annotation agreement between GPT-3.5-turbo and GPT-4-1106-preview.

Models	Coh.	Con.	Flu.	Rel.
hDe-En	0.664	0.631	0.751	0.738
hEn-De	0.729	0.661	0.770	0.748

to references according to our four evaluation criteria, namely coherence, consistency, fluency, and relevance.

Machine-generated summaries are different from human-written summaries in that the generated summaries: (1) lack fluency (except for GPT4), containing grammar errors, punctuation errors, and untranslated texts as shown in Table 12. The automatic summaries sometimes contain nonsensical sentences such as *"Once again, hedge and hedge become hedges again."* (from mLED-Base with fluency of 1); (2) contain factual inconsistencies, especially in supervised abstractive models. For example, a generated summary from mLED-Base (supervised abstractive model) with relevance of 1 starts with *"A rich man has two daughters. The first is beautiful, the second is beautiful and the third is beautiful."* which greatly contradicts the source and reference. Indeed, the human-written reference mentions "only daughter" in the very first sentence. Moreover, the generated sentence by itself is also inconsistent concerning the number of daughters; (3) lack relevance. Sometimes, the generated summaries omit important plots provided in human-written references and the summaries may become overly concise in this case. For example, ChatGPT summarizes that *"Her father remarries and her stepmother and stepsisters treat her cruelly. With the help of some birds, cinderella is able to attend a ball and dance with the prince."* while the human-written reference contains the description of the event: *"The king decides to proclaim a festival that will last for three days and invites all the beautiful maidens in that country to attend so that the prince can select one of them for his bride."* Without this information, the understandability of the story drops and such omissions lead to points reduction in relevance and coherence; (4) lack coherence, especially for abstractive models. There are also cases where the texts are incomprehensible. For example, here is one generation with coherence of 1: *"[...] He asks him where he was when he was at the castle [...] The third asks what he learned, and he explains the reason he did so [...] He goes to church and admonishes the giants and flies. He follows them until the world becomes filled with birds, and then the world is filled with flies and spider-smugglingbirds."* (5) have different distributions of lengths compared to the references depending on the models. Outputs from abstractive models usually contain fewer tokens than human-written references, on average 27 tokens less for supervised abstractive models and more than 100 tokens less for zero-shot abstractive models. This also explains why zero-shot models omit plots. In contrast, the generated outputs from extractive models contain 300 more tokens on average and usually contain more information than the reference, including more details or even redundant contents such as full conversations and repeated information.

8.8 Details on Regression Analysis

8.8.1 Variance Inflation Factors (VIF). Table 23 contains the VIF for numerical features.

Table 23
Variance inflation factor for numerical features. The VIF for the publication year is computed based on the numerical value of the year.

Feature	hDe-En	hEn-De
Publication year	1.04	2.45
Length of document	1.27	2.60
Mean dependency distance	1.02	2.19
Embedding similarity	1.30	1.55

Table 24
Regression results for model features.

Model	hDe-En	hEn-De
	Supervised Extractive	
MemSum		
-translation$_{\text{max 25}}$	-0.06	-0.94^{***}
-Norma-translation$_{\text{max 25}}$	-0.19^{**}	-0.98^{***}
	Supervised Abstractive - Baseline CLCTS	
mLED		
-Historical MT	-0.0	0.08
-Bidirection$_{\text{Prefix}}$	0.08	0.08
	Supervised Abstractive - intermediate fine-tuning	
mLED		
-MLS$_{\text{tgt}}$	-0.01	0.02
-MLS$_{\text{src+tgt}}$	-0.13^{*}	-0.07
-MLS-CLS	-0.22^{***}	0.09
-MLS-CLS-CTS	-0.16^{**}	0.08
	Zero-shot Abstractive - ChatGPT	
ChatGPT as summarizer		
-e2e (Title) prompt	-0.37^{***}	0.06
-e2e prompt	1.47^{***}	0.64^{***}
-pipeline prompt	1.30^{***}	0.59^{***}
-retrieve-ChatGPT$_{\text{max 100}}$	1.30^{***}	0.55^{***}

8.8.2 Regression Coefficient for Variable Model. In Table 24 we list the regression coefficient for the variable *model*.

8.9 Examples of Adversarial Attacks

We list here one example from a short story with all three attacks. All successful cases are highlighted in green and the failures in red. We also highlight the swapped entities in orange in the adversarial text with the corresponding entities in the source text in *italics*.

8.9.1 Source Text. "A little brother and sister were once playing by a *well*, and while they were thus playing, they both fell in. **A water-nix** lived down below, who said, "Now I have got you, now you shall work hard for me!" and carried them off with her. She gave the girl dirty tangled flax to spin, and she had to fetch *water* in a bucket with a hole in it, and the boy had to hew down a tree with a blunt axe, and they got nothing to eat but dumplings as hard as stones. Then at last the children became so impatient, that they waited until one Sunday, when the nix was at church, and ran away. But when church was over, the nix saw that the birds were flown, and followed them with great strides. The children saw her from afar, and the girl threw a *brush* behind her which formed an immense *hill of bristles*, with thousands and thousands of spikes, over which the nix was forced to scramble with great difficulty; at last, however, she got over. When the children saw this, the boy threw behind him a *comb* which made a great *hill of combs* with a thousand times a thousand teeth, but the nix managed to keep herself steady on them, and at last crossed over that. Then the girl threw behind her a looking-glass which formed a hill of mirrors, and was so slippery that it was impossible for the nix to cross it. Then she thought, "I will go home quickly and fetch my *axe*, and cut the hill of glass in half." Long before she returned, however, and had hewn through the glass, the children had escaped to a great distance, and the water-nix was obliged to betake herself to her well again."

8.9.2 Entity Swap. "A little brother and sister were once playing by a cave, and while they were thus playing, they both fell in. A water-nix lived down below, who said, "Now I have got you, now you shall work hard for me!" and carried them off with her. She gave the girl dirty tangled flax to spin, and she had to fetch wine in a bucket with a hole in it, and the boy had to hew down a tree with a blunt axe, and they got nothing to eat but dumplings as hard as stones. Then at last the children became so impatient, that they waited until one Sunday, when the nix was at church, and ran away. But when church was over, the nix saw that the birds were flown, and followed them with great strides. The children saw her from afar, and the girl threw a durian behind her which formed an immense hill of durians, with thousands and thousands of spikes, over which the nix was forced to scramble with great difficulty; at last, however, she got over. When the children saw this, the boy threw behind him a pineapple which made a great hill of pineapples with a thousand times a thousand spikes, but the nix managed to keep herself steady on them, and at last crossed over that. Then the girl threw behind her a looking-glass which formed a hill of mirrors, and was so slippery that it was impossible for the nix to cross it. Then she thought, "I will go home quickly and fetch my butcher knife, and cut the hill of glass in half." Long before she returned, however, and had hewn through the glass, the children had escaped to a great distance, and the water-nix was obliged to betake herself to her well again."

Successful case from ChatGPT: "A brother and sister were playing near a cave when they fell in. They were captured by a water nymph who made them work hard and gave them only hard dumplings to eat. The children waited until the nymph went to church on a Sunday and then escaped. The nymph chased them, but the children threw prickly fruits and a mirror at her, causing her to struggle to climb over them. The children managed to escape while the nymph went to get a tool to break the mirror."

Failure Case from ChatGPT: "A brother and sister were playing near a cave and fell in. They were captured by a water nymph who made them work hard, giving the girl bad flax to spin and making the boy chop a tree with a blunt axe, while only feeding them hard dumplings. One Sunday, they managed to escape when the nymph went to church. The nymph chased them, but the children threw different fruit with spiky exteriors behind them, making it hard for the nymph to catch up. Finally, the girl threw a smooth mirror, forcing the nymph to go home for an axe to break through it. By the time she returned, the children had already escaped."

8.9.3 Negation. "**A water-nix** was once playing by a well, and while it was thus playing, it fell in. **A little brother and sister** lived down below, who said, "Now we have got you, now you shall work hard for **us**!" and carried it off with **them**. **They gave the nix** dirty tangled flax to spin, and **the water-nix had to** fetch water in a bucket with a hole in it, and **the water-nix had to also** hew down a tree with a blunt axe, and **the water-nix got nothing to eat** but dumplings as hard as stones. Then at last it became so impatient, that it waited until one Sunday, when **the brother and sister were at church**, and ran away. But when church was over, **they** saw that the birds were flown, and followed them with great strides. The **water-nix saw them** from afar, and threw a brush behind her which formed an immense hill of bristles, with thousands and thousands of spikes, over which they were forced to scramble with great difficulty; at last, however, **they got over**. When **the nix saw** this, **the nix threw** behind it a comb which made a great hill of combs with a thousand times a thousand teeth, but they managed to keep themselves steady on the combs, and at last crossed over that. Then **the nix threw** behind her a looking-glass which formed a hill of mirrors, and was so slippery that it was impossible for them to cross it. Then they thought, "**We will go** home quickly and fetch our axes, and cut the hill of glass in half." Long before they returned, however, and had hewn through the glass, **the nix had escaped to a great distance, and the brother and sister were obliged to betake themselves to their well again**."

Successful Case from ChatGPT: "A water-nix fell into a well and was captured by a brother and sister who made it work hard for them. The water-nix had to spin flax, fetch water with a leaking bucket, chop down a tree with a blunt axe, and eat hard dumplings. The water-nix escaped when the siblings went to church, but they chased after it. The water-nix threw obstacles in their way, including a hill of bristles, a hill of combs, and a hill of mirrors. The siblings managed to cross the first two hills but were unable to cross the hill of mirrors. They decided to go home and get their axes to cut the hill in half, but by the time they returned, the water-nix had escaped."

Failure Case from ChatGPT: "A water-nix falls into a well and is captured by a brother and sister who force it to spin dirty flax, fetch water in a leaky bucket, and do other difficult tasks. The water-nix escapes on a Sunday when the siblings are at church, but they follow and the water-nix hinders them with a hill of bristles, a hill of combs, and a hill of mirrors. They plan to cut the glass hill in half but the nix escapes before they can return with their axes. The story ends with the water-nix back in the well and the brother and sister empty-handed."

8.9.4 Omission. "A little brother and sister were once playing by a well, and while they were thus playing, they both fell in. A water-nix lived down below, who said, "Now I have got you, now you shall work hard for me!" and carried them off with her. She gave the girl dirty tangled flax to spin, and she had to fetch water in a bucket with a hole in it, and the boy had to hew down a tree with a blunt axe, and they got nothing to eat but dumplings as hard as stones. Then at last the children became so impatient, that they waited until one Sunday, when the nix was at church, and ran away. But when church was over, the nix saw that the birds were flown, and followed them with great strides. The children saw her from afar, and the girl threw a brush behind her which formed an immense hill of bristles, with thousands and thousands of spikes, over which the nix was forced to scramble with great difficulty; at last, however, she got over. When the children saw this, the boy threw behind him a comb which made a great hill of combs with a thousand times a thousand teeth, but the nix managed to keep herself steady on them, and at last crossed over that. Then the girl threw behind her a looking-glass which formed a hill of mirrors, and was so slippery that it was impossible for the nix to cross it. Then she thought, “I will go home quickly and fetch my axe, and cut the hill of glass in half.”

Successful Case from ChatGPT: “A brother and sister fall into a well and are taken by a water-nix. They are forced to work hard and eat only hard dumplings. One Sunday when the nix is at church, the children run away. The nix chases them, but the children throw objects that create obstacles for her to cross. The nix eventually crosses them all except for a hill of mirrors. She decides to go home and fetch her axe to cut the hill in half.”

Failure Case from ChatGPT: “A brother and sister were playing by a well when they both fell in and were captured by a water nymph. The nymph made the girl spin flax and carry water while the boy had to chop a tree with a blunt axe. They had nothing to eat but hard dumplings. They waited for the nymph to go to church one Sunday and then fled. The nymph chased them but they threw a brush, a comb, and a mirror behind them which turned into impenetrable obstacles. The nymph eventually gave up and the children escaped.”

8.10 Examples of Failure Cases for Adversarial Attacks

Tables 25, 26, 27 show examples of failure cases for adversarial attacks concerning omission, entity swap, and negation, respectively.

Omission

Table 25
Example of failure cases of omission attack. The failure is highlighted in red.

Attack details
In the story of Cinderella (“Aschenputtel” in German), we omit the last part of the story where Cinderella attends the ball and meets the prince.

Generated outputs
“Once upon a time, there was a wealthy man who lived happily with his wife and their daughter. When the wife became sick and died, she instructed her daughter to plant a tree on her grave and promised to help her if she shook the tree when in need. The man remarried and his new wife treated the daughter poorly, making her work in the kitchen and giving her the nickname “Cinderella”. When the King held a ball to find a wife for his son, Cinderella’s stepmother made her stay home. However, with the help of two doves, Cinderella was able to attend the ball and see her stepmother dancing with the Prince. The next day, the stepmother was angry that Cinderella had read the lentils and couldn’t scold her.”

Entity swap

Table 26
Example of failure cases of entity swap attack. The successful swap is heightened in green and the failure is highlighted in red.

Attack details
In this story, we replace (1) the well (which the siblings fall into) –> cave; (2) comb and brushes (with which the siblings use to create obstacles) –> pineapples and durian; (3) the axe (which the nymph fetches) –> butcher knife.

Generated outputs
“A brother and sister were playing near a cave and fell in. They were captured by a water nymph who made them work hard, giving the girl bad flax to spin and making the boy chop a tree with a blunt axe, while only feeding them hard dumplings. One Sunday, they managed to escape when the nymph went to church. The nymph chased them, but the children threw different fruit with spiky exteriors behind them, making it hard for the nymph to catch up. Finally, the girl threw a smooth mirror, forcing the nymph to go home for an axe to break through it. By the time she returned, the children had already escaped.”

Negation

Table 27
Example of failure cases of negation attack. The failure is highlighted in red.

Attack details
In the story of the frog prince ("Der Froschkönig oder der eiserne Heinrich" in German), we negate the story by drastically changing the ending where the princess refuses to keep her promise and kills the frog.

Generated outputs
"Once upon a time, a princess was playing with her golden ball when it fell into a deep well. She cried and wished for someone to retrieve it for her. A frog offered to help her in exchange for becoming her companion and eating from her plate and sleeping in her bed. The princess agreed and the frog retrieved the ball. However, when the frog came to claim his reward, the princess was scared and refused to keep her promise. The king then made her keep her word and the frog turned into a prince."

Acknowledgments

We thank all annotators for their hard work and time. The NLLG group gratefully acknowledges support from the Federal Ministry of Education and Research (BMBF) via the grant "Metrics4NLG" and the German Research Foundation (DFG) via the Heisenberg grant EG375/5-1.

References

Adesam, Yvonne, Malin Ahlberg, and Gerlof Bouma. 2012. bokstaffua, bokstaffwa, bokstafwa, bokstaua, bokstawa... towards lexical link-up for a corpus of old Swedish. In *KONVENS*, pages 365–369.

Agarwal, Divya, Devika Vijay, et al. 2021. Genre classification using character networks. In *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 216–222.

Bai, Yu, Yang Gao, and Heyan Huang. 2021. Cross-lingual abstractive summarization with limited parallel resources. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6910–6924. https://doi.org/10.18653/v1/2021.acl-long.538

Bai, Yu, Heyan Huang, Kai Fan, Yang Gao, Yiming Zhu, Jiaao Zhan, Zewen Chi, and Boxing Chen. 2022. Unifying cross-lingual summarization and machine translation with compression rate. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1087–1097. https://doi.org/10.1145/3477495.3532071

Balloccu, Simone, Patrícia Schmidtová, Mateusz Lango, and Ondrej Dusek. 2024. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source LLMs. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 67–93.

Bang, Yejin, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–718. https://doi.org/10.18653/v1/2023.ijcnlp-main.45

Baron, Alistair and Paul Rayson. 2008. Vard2: A tool for dealing with spelling variation in historical corpora. In *Postgraduate Conference in Corpus Linguistics*. https://eprints.lancs.ac.uk/id/eprint/41666/

Belouadi, Jonas and Steffen Eger. 2023. UScore: An effective approach to fully unsupervised evaluation metrics for machine translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 358–374. https://doi.org/10.18653/v1/2023.eacl-main.27

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

Belz, Anya, Anastasia Shimorina, Shubham Agarwal, and Ehud Reiter. 2021. The ReproGen shared task on reproducibility of human evaluations in NLG: Overview and results. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 249–258, Association for

Computational Linguistics. https://doi.org/10.18653/v1/2021.inlg-1.24

Belz, Anya, Craig Thomson, and Ehud Reiter. 2023. Missing information, unresponsive authors, experimental flaws: The impossibility of assessing the reproducibility of previous human evaluations in NLP. In *The Fourth Workshop on Insights from Negative Results in NLP*, pages 1–10. https://doi.org/10.18653/v1/2023.insights-1.1

Bertsch, Amanda, Uri Alon, Graham Neubig, and Matthew Gormley. 2024. Unlimiformer: Long-range transformers with unlimited length input. *Advances in Neural Information Processing Systems*, 36.

Bhandari, Manik, Pranav Narayan Gour, Atabak Ashfaq, Pengfei Liu, and Graham Neubig. 2020. Re-evaluating evaluation in text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9347–9359. https://doi.org/10.18653/v1/2020.emnlp-main.751

Bollmann, Marcel. 2012. Automatic normalization of historical texts using distance measures and the Norma tool. In *Proceedings of the Second Workshop on Annotation of Corpora for Research in the Humanities (ACRH-2)*, pages 3–14.

Bollmann, Marcel. 2019. A large-scale comparison of historical text normalization systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3885–3898.

Bollmann, Marcel, Joachim Bingel, and Anders Søgaard. 2017. Learning attention for historical text normalization by learning to pronounce. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 332–344. https://doi.org/10.18653/v1/P17-1031

Bollmann, Marcel and Anders Søgaard. 2016. Improving historical spelling normalization with bi-directional LSTMs and multi-task learning. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 131–139.

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Cao, Yue, Hui Liu, and Xiaojun Wan. 2020. Jointly learning to align and summarize for neural cross-lingual summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6220–6231. https://doi.org/10.18653/v1/2020.acl-main.554

Chang, Ting Yun and Chi-Jen Lu. 2021. Rethinking why intermediate-task fine-tuning works. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 706–713. https://doi.org/10.18653/v1/2021.findings-emnlp.61

Chen, Wang, Piji Li, and Irwin King. 2021. A training-free and reference-free summarization evaluation metric via centrality-weighted relevance and self-referenced redundancy. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 404–414. https://doi.org/10.18653/v1/2021.acl-long.34

Chen, Yanran, Jonas Belouadi, and Steffen Eger. 2022. Reproducibility issues for BERT-based evaluation metrics. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2965–2989. https://doi.org/10.18653/v1/2022.emnlp-main.192

Chen, Yanran and Steffen Eger. 2023. Menli: Robust evaluation metrics from natural language inference. *Transactions of the Association for Computational Linguistics*, 11:804–825. https://doi.org/10.1162/tacl_a_00576

Chiang, Cheng Han and Hung-Yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631. https://doi.org/10.18653/v1/2023.acl-long.870

Cui, Peng and Le Hu. 2021. Sliding selector network with dynamic memory for extractive summarization of long documents. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5881–5891. https://doi.org/10.18653/v1/2021.naacl-main.470

Ding, Jiayu, Shuming Ma, Li Dong, Xingxing Zhang, Shaohan Huang, Wenhui Wang, and Furu Wei. 2023. LongNet: Scaling transformers to 1,000,000,000 tokens. *arXiv preprint arXiv:2307.02486*.

Eger, Steffen and Alexander Mehler. 2016. On the linearity of semantic change: Investigating meaning variation via dynamic graph models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 52–58. `https://doi.org/10.18653/v1/P16-2009`

Eger, Steffen, Tim vor der Brück, and Alexander Mehler. 2016. A comparison of four character-level string-to-string translation models for (ocr) spelling error correction. *The Prague Bulletin of Mathematical Linguistics*, 105(1):77. `https://doi.org/10.1515/pralin-2016-0004`

Fabbri, Alexander R., Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409. `https://doi.org/10.1162/tacl_a_00373`

Fan, Angela, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2021. Beyond English-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(1):4839–4886.

Gao, Mingqi, Jie Ruan, Renliang Sun, Xunjian Yin, Shiping Yang, and Xiaojun Wan. 2023. Human-like summarization evaluation with ChatGPT. *arXiv preprint arXiv:2304.02554*.

Gao, Yang, Wei Zhao, and Steffen Eger. 2020. SUPERT: Towards new frontiers in unsupervised evaluation metrics for multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1347–1354. `https://doi.org/10.18653/v1/2020.acl-main.124`

Gibson, Edward, Richard Futrell, Steven P. Piantadosi, Isabelle Dautriche, Kyle Mahowald, Leon Bergen, and Roger Levy. 2019. How efficiency shapes human language. *Trends in Cognitive Sciences*, 23(5):389–407. `https://doi.org/10.1016/j.tics.2019.02.003`, PubMed: 31006626

Gidiotis, Alexios and Grigorios Tsoumakas. 2020. A divide-and-conquer approach to the summarization of long documents. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:3029–3040. `https://doi.org/10.1109/TASLP.2020.3037401`

Giulianelli, Mario, Marco Del Tredici, and Raquel Fernández. 2020. Analysing lexical semantic change with contextualised word representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3960–3973. `https://doi.org/10.18653/v1/2020.acl-main.365`

Goyal, Tanya, Junyi Jessy Li, and Greg Durrett. 2022a. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*.

Goyal, Tanya, Junyi Jessy Li, and Greg Durrett. 2022b. SNaC: Coherence error detection for narrative summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 444–463. `https://doi.org/10.18653/v1/2022.emnlp-main.29`

Grusky, Max, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719. `https://doi.org/10.18653/v1/N18-1065`

Gu, Nianlong, Elliott Ash, and Richard Hahnloser. 2022. MemSum: Extractive summarization of long documents using multi-step episodic Markov decision processes. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6507–6522. `https://doi.org/10.18653/v1/2022.acl-long.450`

Hamilton, William L., Jure Leskovec, and Dan Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1501. `https://doi.org/10.18653/v1/P16-1141`

He, Pengcheng, Baolin Peng, Song Wang, Yang Liu, Ruochen Xu, Hany Hassan, Yu Shi, Chenguang Zhu, Wayne Xiong, Michael Zeng, Jianfeng Gao, and Xuedong Huang. 2023. Z-code++: A pre-trained language model optimized for abstractive summarization. In *Proceedings of the 61st*

Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5095–5112. https://doi.org/10.18653/v1/2023.acl-long.279

Huang, Luyang, 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. https://doi.org/10.18653/v1/2021.naacl-main.112

Ji, Ziwei, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38. https://doi.org/10.1145/3571730

Joseph, Brian D. 2017. Diachronic morphology. *The Handbook of Morphology*, pages 349–373. https://doi.org/10.1002/9781405166348.ch18

Juzek, Tom S., Marie-Pauline Krielke, and Elke Teich. 2020. Exploring diachronic syntactic shifts with dependency length: The case of scientific English. In *Proceedings of the Fourth Workshop on Universal Dependencies (UDW 2020)*, pages 109–119.

Karpinska, Marzena and Mohit Iyyer. 2023. Large language models effectively leverage document-level context for literary translation, but critical errors persist. In *Proceedings of the Eighth Conference on Machine Translation*, pages 419–451. https://doi.org/10.18653/v1/2023.wmt-1.41

Koh, Huan Yee, Jiaxin Ju, Ming Liu, and Shirui Pan. 2022. An empirical survey on long document summarization: Datasets, models, and metrics. *ACM Computing Surveys*, 55(8):1–35. https://doi.org/10.1145/3545176

Laban, Philippe, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177. https://doi.org/10.1162/tacl_a_00453

Ladhak, Faisal, Esin Durmus, Claire Cardie, and Kathleen McKeown. 2020. WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4034–4048. https://doi.org/10.18653/v1/2020.findings-emnlp.360

Lei, Lei and Ju Wen. 2020. Is dependency distance experiencing a process of minimization? A diachronic study based on the State of the Union addresses. *Lingua*, 239:102762. https://doi.org/10.1016/j.lingua.2019.102762

Leiter, Christoph, Ran Zhang, Yanran Chen, Jonas Belouadi, Daniil Larionov, Vivian Fresen, and Steffen Eger. 2024. ChatGPT: A meta-analysis after 2.5 months. *Machine Learning with Applications*, 16:100541. https://doi.org/10.1016/j.mlwa.2024.100541

Lewis, Edwin Herbert. 1894. *The History of the English Paragraph*. 2. AMS Press.

Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703

Liang, Yunlong, Fandong Meng, Chulun Zhou, Jinan Xu, Yufeng Chen, Jinsong Su, and Jie Zhou. 2022. A variational hierarchical model for neural cross-lingual summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2088–2099. https://doi.org/10.18653/v1/2022.acl-long.148

Lin, Chin Yew. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

Liu, Haitao. 2008. Dependency distance as a metric of language comprehension difficulty. *Journal of Cognitive Science*, 9(2):159–191. https://doi.org/10.17791/jcs.2008.9.2.159

Liu, Haitao, Chunshan Xu, and Junying Liang. 2017. Dependency distance: A new perspective on syntactic patterns in natural languages. *Physics of Life Reviews*, 21:171–193. https://doi.org/10.1016/j.plrev.2017.03.002, PubMed: 28624589

Liu, Yang, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment, Bouamor, Houda, Juan Pino, and Kalika Bali, editors.

In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522. https://doi.org/10.18653/v1/2023.emnlp-main.153

Liu, Yizhu, Qi Jia, and Kenny Zhu. 2022. Reference-free summarization evaluation via semantic correlation and compression ratio. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2109–2115. https://doi.org/10.18653/v1/2022.naacl-main.153

Liu, Yixin, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. BRIO: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903. https://doi.org/10.18653/v1/2022.acl-long.207

Makarov, Peter and Simon Clematide. 2020. Semi-supervised contextual historical text normalization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7284–7295. https://doi.org/10.18653/v1/2020.acl-main.650

Manakul, Potsawee and Mark Gales. 2021. Long-span summarization via local attention and content selection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6026–6041. https://doi.org/10.18653/v1/2021.acl-long.470

Niwattanakul, Suphakit, Jatsada Singthongchai, Ekkachai Naenudorn, and Supachanun Wanapu. 2013. Using of Jaccard Coefficient for keywords similarity. In *Proceedings of the International Multiconference of Engineers and Computer Scientists*, volume 1, pages 380–384.

Ouyang, Jessica, Boya Song, and Kathy McKeown. 2019. A robust abstractive system for cross-lingual summarization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2025–2031. https://doi.org/10.18653/v1/N19-1204

Peng, Xutan, Yi Zheng, Chenghua Lin, and Advaith Siddharthan. 2021. Summarising historical text in modern languages. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3123–3142. https://doi.org/10.18653/v1/2021.eacl-main.273

Pettersson, Eva. 2016. *Spelling Normalisation and Linguistic Analysis of Historical Text for Information Extraction*. Ph.D. thesis, Acta Universitatis Upsaliensis.

Peyrard, Maxime. 2019. Studying summarization evaluation metrics in the appropriate scoring range. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5093–5100. https://doi.org/10.18653/v1/P19-1502

Pilault, Jonathan, Raymond Li, Sandeep Subramanian, and Chris Pal. 2020. On extractive and abstractive neural document summarization with transformer language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9308–9319. https://doi.org/10.18653/v1/2020.emnlp-main.748

Popović, Maja. 2015. chrF: Character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395. https://doi.org/10.18653/v1/W15-3049

Porta, Jordi, José-Luis Sancho, and Javier Gómez. 2013. Edit transducers for spelling variation in old spanish. In *Proceedings of the Workshop on Computational Historical Linguistics at NODALIDA 2013. NEALT Proc. Series*, volume 18, pages 70–79.

Qi, Peng, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108. https://doi.org/10.18653/v1/2020.acl-demos.14

Ravaut, Mathieu, Shafiq Joty, and Nancy Chen. 2022. SummaReranker: A multi-task mixture-of-experts re-ranking framework for abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4504–4524. https://doi.org/10.18653/v1/2022.acl-long.309

Rayson, Paul, Dawn Archer, and Nicholas Smith. 2005. Vard versus word: A comparison of the ucrel variant detector and modern spellcheckers on english historical corpora. *Corpus Linguistics 2005*.

Reimers, Nils and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525. https://doi.org/10.18653/v1/2020.emnlp-main.365

Robertson, Alexander and Sharon Goldwater. 2018. Evaluating historical text normalization systems: How well do they generalize? 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 720–725. https://doi.org/10.18653/v1/N18-2113

Rudnicka, Karolina. 2018. Variation of sentence length across time and genre. *Diachronic Corpora, Genre, and Language Change*. 220–240. https://doi.org/10.1075/scl.85.10rud

Sainz, Oscar, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10776–10787. https://doi.org/10.18653/v1/2023.findings-emnlp.722

Saraswat, Mala and Srishti. 2022. Leveraging genre classification with RNN for book recommendation. *International Journal of Information Technology*, 14(7):3751–3756. https://doi.org/10.1007/s41870-022-00937-6

Scialom, Thomas, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2020. MLSUM: The multilingual summarization corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8051–8067. https://doi.org/10.18653/v1/2020.emnlp-main.647

Seabold, Skipper and Josef Perktold. 2010. Statsmodels: Econometric and statistical modeling with python. In *Proceedings of the Python in Science Conference*, page 57. https://doi.org/10.25080/Majora-92bf1922-011

Shen, Chenhui, Liying Cheng, Xuan-Phi Nguyen, Yang You, and Lidong Bing. 2023. Large language models are not yet human-level evaluators for abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4215–4233. https://doi.org/10.18653/v1/2023.findings-emnlp.278

Sherman, Lucius Adelno. 1893. *Analytics of Literature: A Manual for the Objective Study of English Prose and Poetry*. Ginn.

Soni, Mayank and Vincent Wade. 2023. Comparing abstractive summaries generated by ChatGPT to real summaries through blinded reviewers and text classification algorithms. *arXiv preprint arXiv:2303.17650*.

Takase, Sho and Naoaki Okazaki. 2022. Multi-task learning for cross-lingual abstractive summarization. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3008–3016.

Tang, Yuqing, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and fine-tuning. *arXiv preprint arXiv:2008.00401*.

Thai, Katherine, Marzena Karpinska, Kalpesh Krishna, Bill Ray, Moira Inghilleri, John Wieting, and Mohit Iyyer. 2022. Exploring document-level literary machine translation with parallel paragraphs from world literature. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9882–9902. https://doi.org/10.18653/v1/2022.emnlp-main.672

Vyas, Yogarshi, Xing Niu, and Marine Carpuat. 2018. Identifying semantic divergences in parallel text without annotations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1503–1515. https://doi.org/10.18653/v1/N18-1136

Wan, Xiaojun, Huiying Li, and Jianguo Xiao. 2010. Cross-language document summarization based on machine translation quality prediction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 917–926.

Wang, Alex, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020. https://doi.org/10.18653/v1/2020.acl-main.450

Wang, Jiaan, Yunlong Liang, Fandong Meng, Beiqi Zou, Zhixu Li, Jianfeng Qu, and Jie Zhou. 2023. Zero-shot cross-lingual summarization via large language models. In *Proceedings of the 4th New Frontiers in Summarization Workshop*, pages 12–23. https://doi.org/10.18653/v1/2023.newsum-1.2

Wang, Yaqin and Haitao Liu. 2017. The effects of genre on dependency distance and dependency direction. *Language Sciences*, 59:135–147 . https://doi.org/10.1016/j.langsci.2016.09.006

Weller, Orion, Kevin Seppi, and Matt Gardner. 2022. When to use multi-task learning vs intermediate fine-tuning for pre-trained encoder transfer learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 272–282. https://doi.org/10.18653/v1/2022.acl-short.30

Yang, Xianjun, Yan Li, Xinlu Zhang, Haifeng Chen, and Wei Cheng. 2023. Exploring the limits of ChatGPT for query or aspect-based text summarization. *arXiv preprint arXiv:2302.08081*.

Yao, Jin ge, Xiaojun Wan, and Jianguo Xiao. 2015. Phrase-based compressive cross-language summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 118–127. https://doi.org/10.18653/v1/D15-1012

Yuan, Weizhe, Graham Neubig, and Pengfei Liu. 2021. BARTScore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277.

Zaheer, Manzil, Guru Guruganesh, Kumar 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. *Advances in Neural Information Processing Systems*, 33:17283–17297.

Zhang, Haopeng, Xiao Liu, and Jiawei Zhang. 2023. Extractive summarization via ChatGPT for faithful summary generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3270–3278. https://doi.org/10.18653/v1/2023.findings-emnlp.214

Zhang, Jingqing, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020a. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339.

Zhang, Tianyi, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

Zhao, Wei, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. 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 563–578. https://doi.org/10.18653/v1/D19-1053

Zhao, Wei, Michael Strube, and Steffen Eger. 2023. DiscoScore: Evaluating text generation with BERT and discourse coherence. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3865–3883. https://doi.org/10.18653/v1/2023.eacl-main.278

Zhu, Haoran, Xueying Liu, and Nana Pang. 2022. Investigating diachronic change in dependency distance of modern English: A genre-specific perspective. *Lingua*, 272:103307. https://doi.org/10.1016/j.lingua.2022.103307

Zhu, Junnan, Qian Wang, Yining Wang, Yu Zhou, Jiajun Zhang, Shaonan Wang, and Chengqing Zong. 2019. NCLS: Neural cross-lingual 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 3054–3064. https://doi.org/10.18653/v1/D19-1302