SUBSUME: A Dataset for Subjective Summary Extraction from Wikipedia Documents

Nishant Yadav[1]*, Matteo Brucato[1]*, Anna Fariha[2]*,
Oscar Yongquist[1], Julian Killingback[1], Alexandra Meliou[1], and Peter J. Haas[1]
[1]College of Information and Computer Sciences, University of Massachusetts Amherst
[2]Microsoft
annafariha@microsoft.com
{nishantyadav, matteo}@cs.umass.edu
{oyoungquist, jkillingback, ameli, phaas}@cs.umass.edu

Abstract

Many applications require generation of summaries tailored to the user's information needs, i.e., their intent. Methods that express intent via explicit user queries fall short when query interpretation is *subjective*. Several datasets exist for summarization with objective intents where, for each document and intent (e.g., "weather"), a single summary suffices for all users. No datasets exist, however, for subjective intents (e.g., "interesting places") where different users will provide different summaries. We present SUBSUME, the first dataset for evaluation of SUBjective SUMmary Extraction systems. SUBSUME contains 2,200 $(document, intent, summary)$ triplets over 48 Wikipedia pages, with ten intents of varying subjectivity, provided by 103 individuals over Mechanical Turk. We demonstrate statistically that the intents in SUBSUME vary systematically in subjectivity. To indicate SUBSUME's usefulness, we explore a collection of baseline algorithms for subjective extractive summarization and show that (i) as expected, example-based approaches better capture subjective intents than query-based ones, and (ii) there is ample scope for improving upon the baseline algorithms, thereby motivating further research on this challenging problem.

1 Introduction

Traditional non-generic extractive summarization systems allow users to express their summarization intent via a query or a natural-language question (Daumé III and Marcu, 2006; Li and Li, 2014; Verberne et al., 2020). While this simplifies the interaction between the user and the system, queries are not the best means for expressing very *subjective* intents. Consider a user trying to summarize the Wikipedia pages of all US states to find places that would be *interesting* to them. A query such as "interesting places" may report places that are of general interest (e.g., interesting in terms of popularity), thus failing to model the subjectiveness of the concept "interesting". Revising the query (e.g., by adding "art museums" or "surfing spots") can be a complex, iterative process, which is frustrating for the user. Instead, we argue that in cases like this, where the user wants to summarize many documents with the same intent, it is often easier to communicate subjective intents by providing *examples* for a few states, from which the system can infer the intent more effectively.

The example-based paradigm *programming-by-example* (PBE) has been successful for a variety of tasks, such as: code synthesis (Drosos et al., 2020); data wrangling (Gulwani, 2016; FlashFill), integration (Inala and Singh, 2017), and extraction (Le and Gulwani, 2014); text processing and normalization (Yessenov et al., 2013; Kini and Gulwani, 2015); querying relational databases (Fariha and Meliou, 2019), and even creative tasks such as music composition (Frid et al., 2020).

An interface for extractive summarization by example was proposed in SUDOCU (Fariha et al., 2020), offering an easy and natural way for users to annotate documents to construct example summaries: the user browses through the document, optionally performing keyword search, and simply clicks on sentences that should be included in the summary. The system then infers the user's intent from the provided examples, and learns the mechanism to automatically summarize the rest of the unseen documents. Figure 1 contrasts the traditional query-based interface (left) with an example-based one (right). The interface makes it easy for users to construct a few example summaries from a corpus.

Summarization by example is powerful for several reasons: First, it allows the system to access more information than what a query might provide, and, thus, such a paradigm is expected to produce better results than the traditional query-based approaches. Second, it allows users to express very

* Equal contribution.

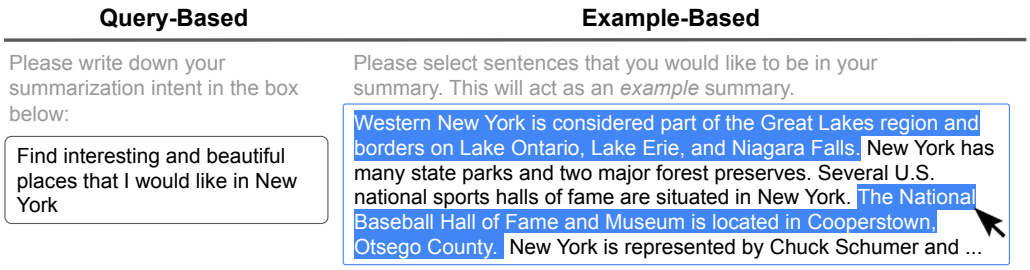

Figure 1: Query-based interface vs. Example-based interface for document summarization.

subjective intents precisely where typical methods fail (e.g., a natural-language query that asks a system to find "places that I like"). Third, it relieves the user from constructing the correct query even for an objective intent: people are often more comfortable in giving a few *examples* of what they want than providing specifications of what they want.

Because an example-based summarization system uses a different input (i.e., the example summaries) than a query-based system (i.e., a query), evaluating an example-based system needs a more complex *evaluation dataset* than those available for query-based systems. Given an intent, we need a few summaries—all produced by the same user—where a subset of the summaries are used as examples and the rest are used to evaluate the summaries that the system produces. Unfortunately, existing summarization datasets provide only one summary per user-intent pair. We present SUBSUME, the first dataset for evaluating SUBjective SUMmary Extraction systems. SUBSUME is suitable for evaluating example-based summarization systems, as it includes 8 different, manually curated summaries, produced by the same user, for every user-intent pair. Further, SUBSUME is the first dataset to include intents with an increasing level of subjectivity. SUBSUME can also be used to evaluate generic (Hong et al., 2014), query-based, question-based, and even abstractive (Nallapati et al., 2016) summarization systems, as the example-driven paradigm *subsumes* them all.

To demonstrate how SUBSUME can be used, we empirically compare several baselines on intents with increasing subjectivity. SUBSUME exposes evidence that (i) as expected, an example-based approach better captures subjective intent than a naive approach that simply inputs an ambiguous intent into a query-based summarizer and (ii) there is ample scope for improving upon the baseline algorithms, thereby motivating further research on this challenging problem.

2 Related Work

Several datasets exist for generic summarization tasks, including the CNN/Daily Mail dataset (Nallapati et al., 2016) which contains 300,000 news article-summary pairs, Webis-TLDR-17, which contains three million document-summary pairs extracted from Reddit forums (Völske et al., 2017), Multi-News dataset, which is a multi-document summarization dataset containing over 50,000 articles-summary pairs (Fabbri et al., 2019), and the Gigaword (Rush et al., 2015) and X-Sum (Narayan et al., 2018) datasets, both of which contain single-sentence summaries of news articles.

ScisummNet (Yasunaga et al., 2019) is a manually annotated corpus for scientific papers on computational linguistics to generate summaries that include the impacts of the articles on the research community. TalkSumm (Lev et al., 2019) is for scientific paper summarization based on conference talks. However, it does not consider personalization, where different people might want different summaries of the same paper. In general, none of the above datasets are suitable for the task of subjective summarization, which is our focus.

A task close to ours is *query* or *topic-based* extractive summarization. Suitable datasets include DUC 2004, DUC 2005, and DUC 2006, which contain query-based (multi-)document summaries (DUC). Webis-Snippet-20 consists of 10M web pages together with their query-based abstractive snippets (Chen et al., 2020). In these datasets, each document (or set of documents) has one or more summaries with respect to a single query. In contrast, SUBSUME contains multiple summaries of each document corresponding to *different intents*. Furthermore, each document-intent pair is summarized by multiple individuals.

Frermann and Klementiev (2019), in the context of "aspect-based" summarization, provide a dataset having multiple topic-focused summaries

	Mostly Objective
(I1)	How is the weather of the state?
(I2)	How is the government structured in this state?
(I3)	What is the state's policy regarding education?
(I4)	What are the available modes of transport in this state?
	Balanced Subjective/Objective
(I5)	What drives the economy in this state?
(I6)	What are the major historical events in this state?
	Mostly Subjective
(I7)	What about this state's arts and culture attracts you the most?
(I8)	Which places seem interesting to you for visiting in this state?
(I9)	What are some of the most interesting things about this state?
(I10)	What are the main reasons why you would like living in this state?

Figure 2: Intents used in the SUBSUME dataset.

for each document. The dataset is synthetic, however, and does not involve human annotators. To the best of our knowledge, SUBSUME is the first human-generated dataset for subjective, extractive document summarization, where interpretation of intents varies across individuals.

3 Dataset Description

We now describe our data collection process and design choices, and analyze the statistical properties of the dataset. The dataset is available publicly at https://github.com/afariha/SubSumE.

Intents. We devised ten intents with different degrees of subjectiveness, ranging from mostly objective to mostly subjective, as shown in Figure 2. "Objective" intents refer to unambiguous facts (weather, modes of transport), "subjective" intents refer to opinions (interesting, attractive), and the "balanced" intents correspond to a ranking of unambiguous facts according to subjectively estimated importance (drivers, major events) that are expected to vary only moderately between individuals. A statistical analysis (see below) supports our heuristic classification of intents.

Documents. As the source documents, we used English Wikipedia pages of 48 U.S. states. We removed Nebraska and Wyoming as their pages did not have enough content with respect to the chosen intents. We parsed the pages to get text content from paragraph tags, and extracted sentences using Punkt sentence tokenizer from the NLTK library (Loper and Bird, 2002). Our corpus includes homogeneous documents to allow summarization of all documents with respect to all intents. In particular, we chose the Wikipedia pages for the states in the USA because they are homogeneous and contain information on a wide range of topics.

Interface. We collected extractive summaries of the documents using a custom interface on Amazon Mechanical Turk (MTurk). Our interface allowed the workers to search the document for keywords, click on a sentence to include it in the summary, and remove a sentence from the summary. A detailed discussion on the interface is in Appendix B.

Task. Each MTurk task (HIT) required a worker to extract sentences from eight documents to best summarize them according to a given intent, resulting in eight $(document, intent, summary)$ triplets. To generate unique HITs, we partitioned the set of 48 documents into six disjoint sets, each containing eight documents. We then paired each of the six sets with each of the ten intents, resulting in 60 unique HITs. We repeated the above procedure five times to obtain a total of 300 HITs. Out of these 300 HITs, 25 were rejected upon manual inspection (due to poor-quality summaries). The remaining 275 HITs contained eight summaries each, resulting in a total of 2,200 $(document, intent, summary)$ triplets. We allowed workers to participate in multiple HITs as long as they were not identical: either the document-set or the intent was different.

Post-task Survey. We conducted a post-task survey where we asked the workers to provide their interpretation of the intent and any strategies they followed for summarizing. Workers also provided optional demographic information: gender, age, US-residency, English proficiency, and occupation (details are in Appendix E).

Quality Control. We screened noisy workers using MTurk's qualification system. We also inspected the summaries using both automated heuristics and manual inspection to filter out sloppy workers and ensured that the summaries are of good quality and reflect the corresponding intent. A human annotator examined each summary and flagged low-quality ones. For example, for the intent "weather of the state", the annotator flagged a summary as low-quality as it did not contain any weather, but arbitrarily chosen sentences. Additionally, we asked each worker for their interpretation of the task to verify if their task understanding was correct, and excluded summaries in case it was not. Details of our screening test and quality-control mechanisms are in Appendices C and D.

Data Format. We provide SUBSUME in a format to support both query-based and example-driven approaches. Each completed HIT gives us the fol-

Statistic	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10
# Summaries	240	216	232	240	232	224	192	200	208	216
Avg. # sent/summary	11.4	12.7	8.6	10.5	10.8	13.7	11.3	9.3	13.4	11.2
Avg. # words/summary	314	285	227	278	288	380	319	274	375	304
Subjectiveness score	22.7	34.2	35.0	35.6	47.4	58.7	55.7	56.9	74.3	73.2

Table 1: SUBSUME statistics across ten intents.

lowing information and contributes to eight data points in SUBSUME: (1) the intent text (one of I1–I10 in Figure 2), (2) one summary for each of the eight documents in the HIT, (3) interpretation of the intent by the worker, (4) description of summarization strategy followed by the worker, (5) the keywords typed in the search box by the worker while selecting sentences, (6) time-stamps indicating when each sentence was added to the summary, (7) percentage of the document the worker viewed, and (8) optional demographic information of the worker. We include an example datapoint from the dataset in Appendix F.

Dataset Analysis. Table 1 shows statistics of the dataset grouped by intents. We quantify the *subjectiveness* of an intent as follows: Let $\mathcal{S}_{i,d}$ be the set of summaries constructed by all different workers for an intent i and document d. We first compute pair-wise ROUGE-L F_1 scores (normalized between 0 and 100) for all pairs of summaries from $\mathcal{S}_{i,d}$. We define $\texttt{Sim}_{i,d}$ as the average of these scores, measuring the similarity of all pairs of summaries for document d and intent i. We define the *subjectiveness score* (inverse of similarity) for intent i using the following formula: $\texttt{Subj}_i = 100 - \frac{\sum_d \texttt{Sim}_{i,d}}{\sum_d 1}$. The higher the subjectiveness score for a given intent, the lower the similarity among summaries for that intent, thus indicating higher subjectiveness. Our classification of intents in Figure 2 aligns well with this subjectiveness score in Table 1. For instance, "How is the weather of the state?" (**I1**) scores the lowest (22.7) and "What are some of the most interesting things about this state?" (**I9**) scores the highest (74.3).

4 Experiments

In this section, we benchmark existing summarization techniques over SUBSUME in two settings: query-based (QB) and example-driven (EX). Recall that for every user-intent pair, SUBSUME consists of summaries of eight documents. In the EX setting, we use summaries of five documents, chosen at random from the eight summaries, as example summaries to *learn* the user's intent, and evaluate on the remaining three documents. In the QB setting, the baselines summarize the documents using only the query (intent text), and we evaluate on the same set of three documents as in the example-driven setting. We repeat this over ten different splits of the eight document-summary pairs, and average out results across all splits, and over all data points. We report F_1 scores of ROUGE-1, ROUGE-2, and ROUGE-L metrics (Lin, 2004) for all the baselines.

4.1 Baselines

We benchmark the following baselines. We refer the reader to Appendix A for detailed descriptions and implementation details.

KEYWORD first extracts keywords from the example summaries or query text, followed by filtering out sentences with less than t_k keywords. Lastly, a summary is constructed using the top-k sentences with respect to TF-IDF scores.

SBERT embeds example summaries (query text) and sentences in test documents using SBERT (Reimers and Gurevych, 2019). It scores each sentence based on its cosine similarity with the average embedding of the example summaries (query text) and computes a summary using top-k sentences in the document.

PEGASUS is a state-of-the-art abstractive summarization model (Zhang et al., 2020) based on transformers (Vaswani et al., 2017). We use the Pegasus model pre-trained on the CNN-DailyMail dataset.

BERTSUMEXT is a state-of-the-art extractive summarization model (Liu and Lapata, 2019). We use the publicly released model pre-trained on the CNN-DailyMail dataset.

SUDOCU (Fariha et al., 2020) is an example-driven summarization approach that models extractive summarization as an integer linear program.

Metric	Example-Driven (EX)					Query-Based (QB)			
	KEYWORD	SBERT	BERTSUMEXT	PEGASUS	SUDOCU	KEYWORD	SBERT	BERTSUMEXT	PEGASUS
ROUGE-1	30.6	53.2	31.6	23.9	33.2	30.4	41.1	21.7	18.2
ROUGE-2	7.3	36.9	21.1	14.5	15.7	9.6	20.8	10.3	7.7
ROUGE-L	16.7	41.0	23.3	18.2	20.6	16.7	27.1	15.8	13.5

Table 2: ROUGE F_1 scores for baseline techniques averaged across ten random example/test summary splits.

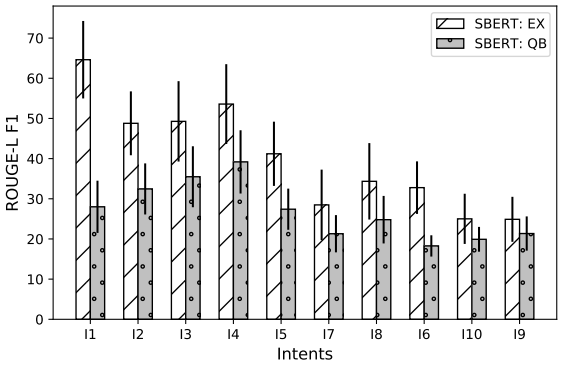

Figure 3: ROUGE-L F_1 for SBERT-EX and SBERT-QB for each intent. From left to right, intents are ordered in increasing order of their subjectiveness score shown in Table 1. The Pearson's correlation between the subjectiveness score and the F_1 score for SBERT-EX and SBERT-QB is -0.97 and -0.77 respectively.

4.2 Results

Table 2 shows the performance of each baseline averaged over all user-intent pairs. As expected, example-driven versions of each baseline consistently outperform their query-based counterparts with SBERT-EX being the top-performing method. This confirms that when users are willing to expend the effort to provide examples, example-driven approaches are superior to query-based ones.

Figure 3 shows the average SBERT ROUGE-L F1-score for each intent in example-driven (EX) and query-based (QB) settings where SBERT-EX consistently outperforms SBERT-QB. As we go from intents with low subjectiveness scores to intents with high subjectiveness scores, the performance of SBERT decreases in both settings. This shows that the task of subjective summarization becomes more challenging with an increase in subjectiveness of the intents.

5 Conclusions

In this paper, we present SUBSUME, the first dataset for evaluation of subjective summarization systems, and evaluate existing baselines on the dataset. The results presented in this paper show that even the best-performing approaches leave significant room for improvement for subjective document summarization, encouraging further research. In future, we plan to investigate transfer-learning and few-shot learning approaches that naturally fit the task of subjective summarization by example. Another direction of future work is to investigate how our data-collection method can be extended to other domains beyond the domain of Wikipedia documents of US states.

6 Ethical Considerations

We obtained Institutional Review Board (IRB) approval for collecting data as it involved human annotators. All workers on MTurk were provided terms and conditions (as approved by IRB) and they could attempt the task only after agreeing to the terms and conditions. A screening test involved answering a few questions in order to filter users based on their English proficiency as the task involved understanding the intent/query and summarizing documents, both of which were in English. In our initial pilots without the screening test, we found that we received noisy datapoints perhaps due to the corresponding worker's limited proficiency in English. More details on the screening task are present in Appendix C. The post-task survey where we requested demographics of the user was completely optional and in no manner affected acceptance or rejection of the task completed by the user. In our initial pilot, we found that most workers could finish the task within an hour. Thus, we decided to pay each worker $6 per HIT, which is typical for an hour-long task. While we collected summaries for Wikipedia pages of the USA states, we believe that our intents and queries could be used to summarize Wikipedia pages of other countries and provinces or states in other countries. We discuss reasons for choosing Wikipedia pages of the states in the USA in Section 3 in more detail.

Acknowledgements

We thank Kanchi Masalia for her help with pilot dataset collection efforts. We would also like to thank anonymous reviewers for their helpful comments and suggestions. This work was supported by the NSF under grants IIS-1453543, IIS-1943971, and CCF-1763423, and a Microsoft Research Dissertation Grant. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

Wei-Fan Chen, Shahbaz Syed, Benno Stein, Matthias Hagen, and Martin Potthast. 2020. Abstractive snippet generation. In *Proceedings of The Web Conference 2020*, pages 1309–1319.

Hal Daumé III and Daniel Marcu. 2006. Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 305–312.

Ian Drosos, Titus Barik, Philip J. Guo, Robert DeLine, and Sumit Gulwani. 2020. Wrex: A unified programming-by-example interaction for synthesizing readable code for data scientists. In *CHI '20: CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, April 25-30, 2020*, pages 1–12.

DUC. 2000-2007. DUC. https://duc.nist.gov/data.html.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084.

Anna Fariha, Matteo Brucato, Peter J. Haas, and Alexandra Meliou. 2020. Sudocu: Summarizing documents by example. *Proc. VLDB Endow.*, 13(12):2861–2864.

Anna Fariha and Alexandra Meliou. 2019. Example-driven query intent discovery: Abductive reasoning using semantic similarity. *Proc. VLDB Endow.*, 12(11):1262–1275.

FlashFill. FlashFill. https://www.microsoft.com/en-us/research/project/flash-fill-excel-feature-office-2013/.

Lea Frermann and Alexandre Klementiev. 2019. Inducing document structure for aspect-based summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6263–6273.

Emma Frid, Celso Gomes, and Zeyu Jin. 2020. Music creation by example. In *CHI '20: CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, April 25-30, 2020*, pages 1–13.

Sumit Gulwani. 2016. Programming by examples - and its applications in data wrangling. In *Dependable Software Systems Engineering*, pages 137–158.

Kai Hong, John M Conroy, Benoit Favre, Alex Kulesza, Hui Lin, and Ani Nenkova. 2014. A repository of state of the art and competitive baseline summaries for generic news summarization. In *LREC*, pages 1608–1616. Citeseer.

Jeevana Priya Inala and Rishabh Singh. 2017. Webrelate: integrating web data with spreadsheets using examples. *Proceedings of the ACM on Programming Languages*, 2(POPL):1–28.

Dileep Kini and Sumit Gulwani. 2015. Flashnormalize: Programming by examples for text normalization. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 776–783.

Vu Le and Sumit Gulwani. 2014. Flashextract: a framework for data extraction by examples. In *ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI '14, Edinburgh, United Kingdom - June 09 - 11, 2014*, pages 542–553.

Guy Lev, Michal Shmueli-Scheuer, Jonathan Herzig, Achiya Jerbi, and David Konopnicki. 2019. Talksumm: A dataset and scalable annotation method for scientific paper summarization based on conference talks. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 2125–2131.

Yanran Li and Sujian Li. 2014. Query-focused multi-document summarization: Combining a topic model with graph-based semi-supervised learning. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1197–1207.

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

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*.

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

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807.

Martin F Porter. 1980. An algorithm for suffix stripping. *Program*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

Suzan Verberne, Emiel Krahmer, Sander Wubben, and Antal van den Bosch. 2020. Query-based summarization of discussion threads. *Natural Language Engineering*, 26(1):3–29.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. TL;DR: Mining Reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R Fabbri, Irene Li, Dan Friedman, and Dragomir R Radev. 2019. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7386–7393.

Kuat Yessenov, Shubham Tulsiani, Aditya Krishna Menon, Robert C. Miller, Sumit Gulwani, Butler W. Lampson, and Adam Kalai. 2013. A colorful approach to text processing by example. In *The 26th Annual ACM Symposium on User Interface Software and Technology, UIST'13, St. Andrews, United Kingdom, October 8-11, 2013*, pages 495–504.

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

Appendix

A Implementation details

For all baselines, we pre-processed the text within the documents by removing all stop-words and converting all characters to lower case; except for SBERT, which does not require stop-words to be removed. For these experiments, we set the value of $k = 10$ for the Keyword-based and SBERT-embeddings-based methods. Additionally, for the Keyword-based approach, we set the keyword coverage threshold t_k to the average keyword-coverage score of all the sentences in the document. Lastly, the number of latent topics extracted by LDA for SUDOCU was limited to 10. For evaluation, we use F_1 scores of the ROUGE-1, ROUGE-2, and ROUGE-L metrics (Lin, 2004) for all the baselines using rouge-score package [1] with Porter stemming (Porter, 1980) turned on.

We implement the following unsupervised baselines, which summarize a new test document based on either the query text (QB) or the example summaries of other documents (EX).

KEYWORD first extracts keywords from the example summaries or query text using Gensim's keywords_extractor method. Each sentence in the test document is then given a *keyword-coverage* score based on the number of keywords it contains. Sentences that cover at least t_k (a threshold) extracted keywords are considered candidate sentences for the summary. Lastly, the TF-IDF scores of the candidate sentences are calculated and the summary is constructed using the top-k sentences; ranked based on their TF-IDF scores.

SBERT embeds example summaries or query text and sentences in test documents using

[1] https://pypi.org/project/rouge-score/

SBERT (Reimers and Gurevych, 2019). It scores each sentence based on its cosine similarity with the average embedding of the example summaries or the query text and computes a summary using top-k high-scoring sentences in the document.

PEGASUS is a state-of-the-art abstractive summarization language model (Zhang et al., 2020). PEGASUS follows the standard Transformer-based encoder-decoder construction popular for abstractive summarization. In this work, we use the HuggingFace API (Wolf et al., 2020) to access a Pegasus model fine-tuned on the CNN-DailyMail dataset released by the authors. In the extractive setting, we use the example summaries or the query text with the pre-trained PEGASUS model as follows. First, the example summaries (or query text) are used to first filter out unimportant sentences before the document is given the PEGASUS to be summarized. This is done by first finding the average SBERT embedding of the example summaries (or query-text). Then, each sentence from the target document is ranked based on their cosine similarity to the average example summary (or query-text) embedding. Finally, the top-$2k$ sentences are used as input for PEGASUS. In the query-driven setting, this same process is performed, but instead of using the example summaries to filter out unimportant sentences, the query itself is used.

BERTSUMEXT is a state-of-the-art extractive summarization model (Liu and Lapata, 2019). BERTSUMEXT introduces a novel, BERT-based document level encoder, which is used as input to several inter-sentence Transformer layers which learn document-level features to guide sentence extraction. In this work we rely on the publicly released models pre-trained on the CNN-DailyMail dataset from the authors of the original work. We use the same pre-filtering approach as used in PEGASUS baseline to use the example summaries or the query text in the summary generation process.

SUDOCU (Fariha et al., 2020) is an example-driven summarization approach that models extractive summarization as an integer linear program.

B HIT Description

Figure 4 shows the instruction page of the data-collection interface and Figure 5 shows the interface where the workers construct the summaries. Below we outline the process the workers go through to complete a HIT.

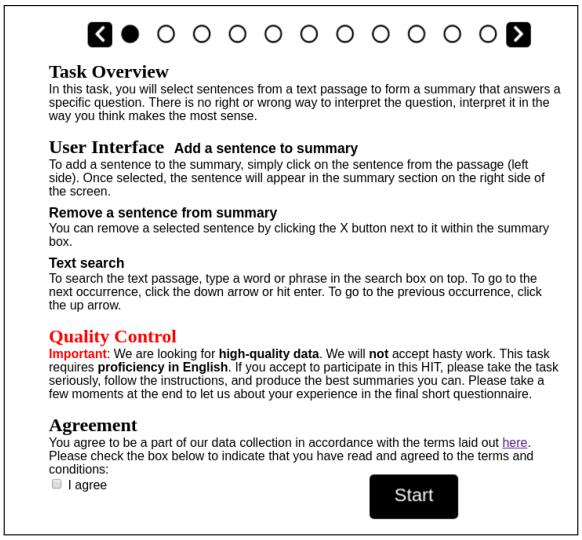

Figure 4: Snapshot of the page with instructions for users on the data collection interface.

1. If a worker has not previously taken our screening test they are prompted to do so, otherwise they can proceed to the task. More information about the screening protocol is in Appendix C,
2. The worker is brought to an instruction page that gives them an overview of the task, instructions on how to use the interface, and our expectations for summaries, and our methods for quality control. We also provide the terms for the study and the worker has to agree to the terms by checking a checkbox before proceeding with the task.
3. After the worker has agreed to the terms and continued to the next page, the summary construction interface is shown for the first document (Figure 5). The intent is shown at the top of the page with the document text directly below it. The worker can search for keywords in the text using the search bar above the document. They can add sentences to the summary by clicking on them and remove them by clicking the "X" button next to the sentence in the summary box (to the right of the document).
4. Once the worker has added the appropriate number of sentences to the summary, they can continue to the next page. We allowed a maximum of 20 sentences in a summary. This process is repeated for eight documents and the user constructs eight summaries in total.
5. On completion of the previous steps, a "summary overview" page (Figure 7) is loaded where the worker can read the summaries for all eight states. If necessary, from this page, they can

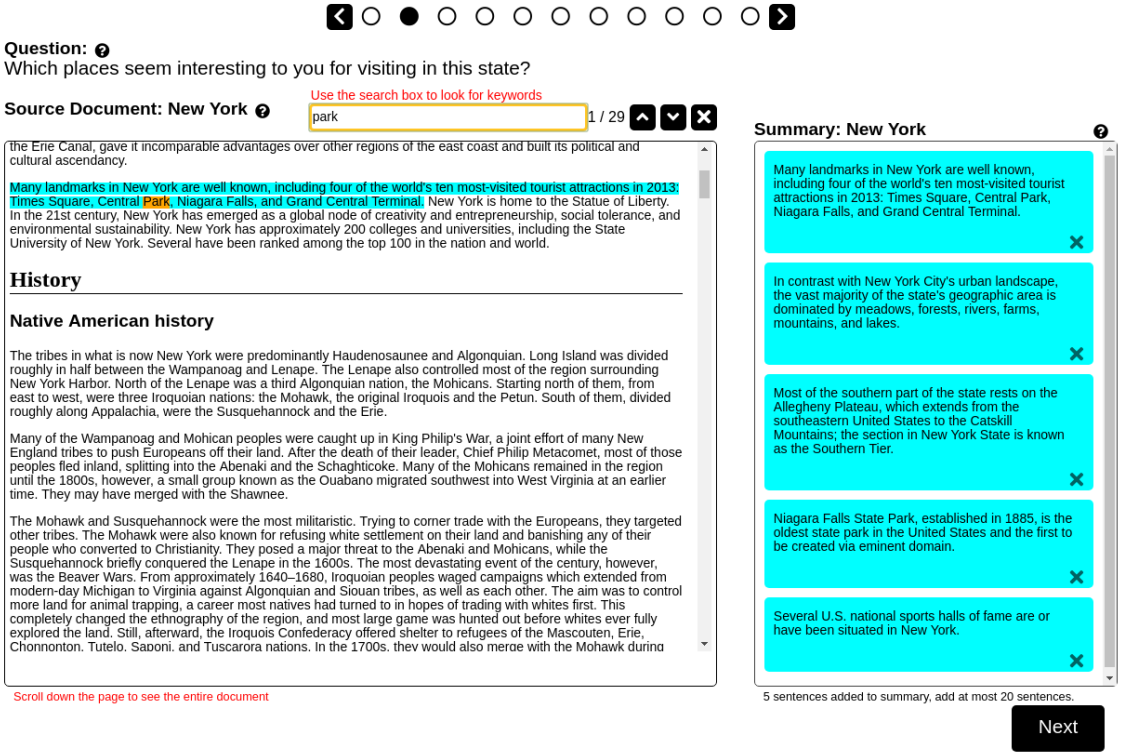

Figure 5: Interface to construct summary used to collect data from workers on Amazon Mechanical Turk.

go back to the individual document pages to modify their summaries.

6. After the worker is satisfied with all the summaries, they submit them, and the task ends.
7. On completion of the task of summary construction, the worker is requested to complete a post-task survey, where they are asked a few questions about their interpretation of intents and strategies they followed to construct the summaries, along with optional questions to collect their demographic information.

C Worker Screening Protocol

During initial pilots, we observed that a large proportion of summaries were of poor quality, generally containing sentences added at random from throughout the document with a majority of sentences having little or no relevance to the intent. To avoid poor-quality summaries, we included a screening test using MTurk's qualification system, and got significantly higher-quality summaries. The screening test consisted of five multiple-choice questions that tested written English skills and knowledge of relevant topics. The first three questions were inspired by the Cambridge English online evaluation test. We created the last two questions manually. Our priority was to make sure workers had a good grasp of written English, as well as to check if they had knowledge of potentially relevant topics such as fiscal policy. A screenshot of the screening test is shown in Figure 6.

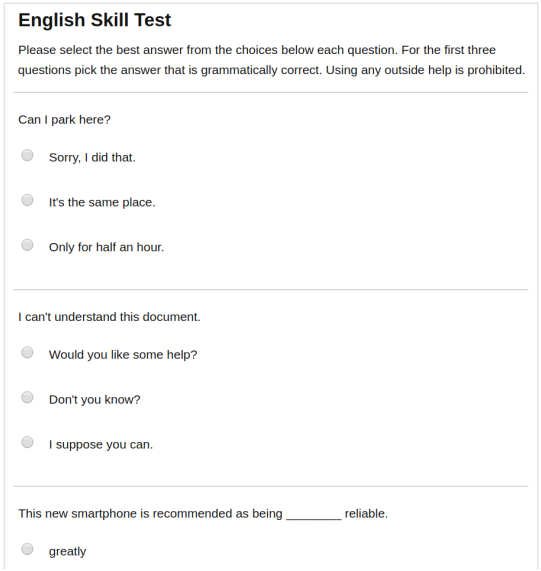

Figure 6: Screenshot of the (partial) screening test workers had to pass before participating in the HITs.

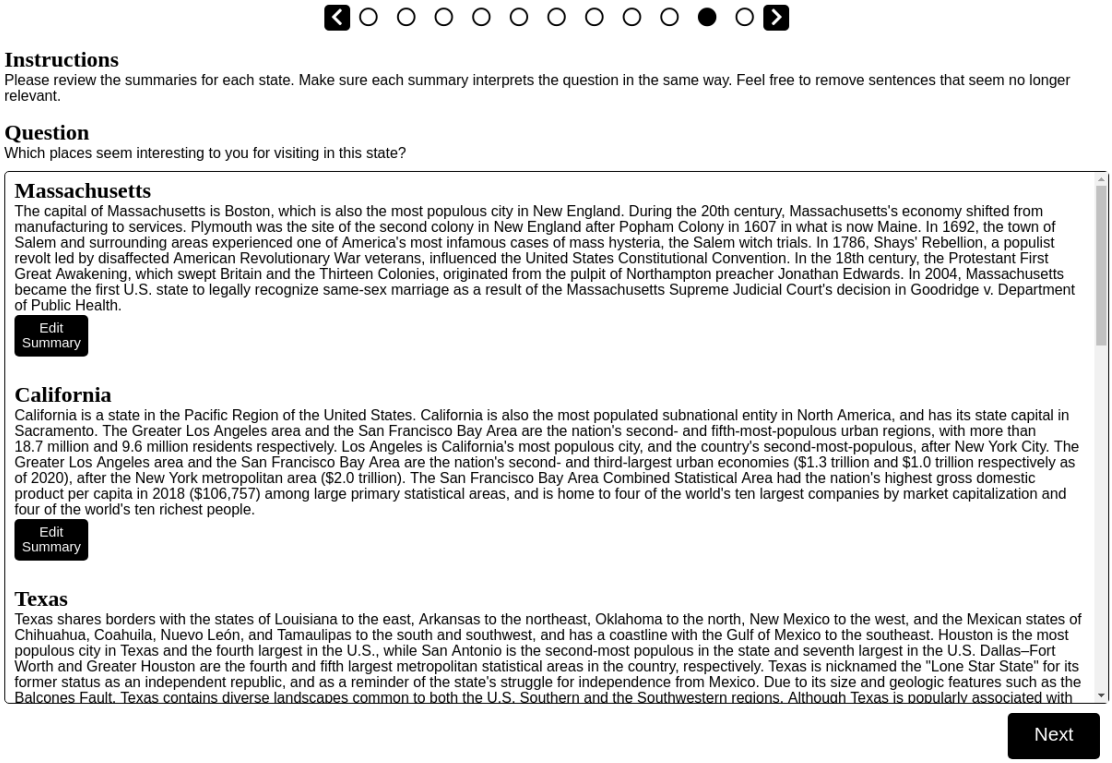

Figure 7: Screenshot of the summary overview page.

D Quality Control

To ensure high-quality summaries, we used a heuristic scoring method to filter out potential poor-quality summaries. Our scoring method is based on the similarity of the distributions of indices of the selected sentences that form the summaries across different documents within a HIT. The intuition is that if someone picks sentences randomly from the documents, then the standard deviation of the indices of the selected sentence are similar across states. This is because if someone picks sentences at random, then it would have a lower standard deviation than when they pick relevant sentences, which are often in various parts of different pages. This scoring works better for mostly objective intents as the sentences related to objective intents are usually more concentrated.

E Worker Demographics

Out of the 103 workers, 47 reported themselves as 'male', 38 as 'female', 16 did not report their gender, and 2 reported 'other'. 36 out of 103 workers are within the age group '30-45', 25 were '16-30', 22 were '45+' years old, and 20 did not report age group. 68 of the workers are US-resident, 13 live outside the U.S., and 22 did not report residency. Finally, 70 out of 103 workers reported very strong English proficiency, while others reported a varying level of English proficiency.

F Example Summaries

A data point, containing one of the 8 summaries within a HIT result, is shown in Figure 8. We omit other information such as time-stamps indicating when each sentence was selected etc for brevity. Figure 9 shows two significantly different summaries of the state of Delaware for the same intent, constructed by two different workers.

Intent: Which places seem interesting to you for visiting in this state?

Summary of Colorado: The northwestern corner of Colorado [. . .] contains part of the noted Dinosaur National Monument, which not only is a paleontological area, but is also a scenic area of rocky hills, canyons, arid desert, and streambeds [. . .] There are also a number of established film festivals in Colorado, including Aspen Shortsfest, Boulder International Film Festival, Castle Rock Film Festival [. . .] The state's diverse geography and majestic mountains attract millions of tourists every year [. . .] Colorado is home to 4 national parks, 8 national monuments, 2 national recreation areas, 2 national historic sites, 3 national historic trails [. . .].

Interpretation: Things I would like to do includes nature spots I would like for outdoor recreation, cities for touring and special events.

Strategy: I looked for the capitol, any other interesting cities, nature, sports, cultural experiences and special events native to the state.

Keywords: tourism, national park, Denver, Mesa

Figure 8: A datapoint from SUBSUME. For brevity, we leave out demographics, document-view-percentage, and timestamp information.

Intent: What are some of the most interesting things about this state?

Summary 1: Delaware: [. . .] The state was named after the Delaware River, which in turn derived its name from Thomas West, 3rd Baron De La Warr (1577–1618) who was the ruling governor of the Colony of Virginia at the time Europeans first explored the river [. . .] A 2012 Gallup poll found that Delaware's proportion of lesbian, gay, bisexual, and transgender adults stood at 3.4 percent of the population [. . .] Gambling provides significant revenue to the state. For instance, the casino at Delaware Park Racetrack provided more than $100 million to the state in 2010 [. . .] Delaware has several museums, wildlife refuges, parks, houses, lighthouses, and other historic places.

Summary 2: Delaware: On December 7, 1787, Delaware became the first state to ratify the Constitution of the United States, and has since been known as The First State. Since the turn of the 20th century, Delaware is also a de facto onshore corporate haven [. . .] The only real engagement on Delaware soil was the Battle of Cooch's Bridge, fought on September 3, 1777, at Cooch's Bridge in New Castle County, although there was a minor Loyalist rebellion in 1778. According to a 2013 study by Phoenix Marketing International, Delaware had the ninth-largest number of millionaires per capita in the United States, with a ratio of 6.2 percent [. . .] Unlike many states, Delaware's educational system is centralized in a state Superintendent of Education, with local school boards retaining control over taxation and some curriculum decisions [. . .] Several ships have been named USS Delaware in honor of this state.

Figure 9: Two example summaries for the Wikipedia page of the state of Delaware for the same intent that demonstrate the range of valid summaries for subjective intents.