SUBSUME: A Dataset for Subjective Summary Extraction from Wikipedia Documents

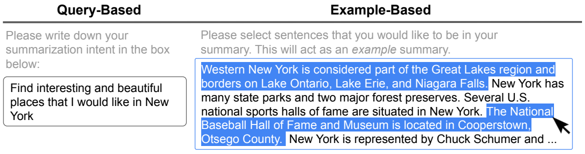
Anonymous EMNLP submission

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 10 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, 2009; 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 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.



Please write down your summarization intent in the box below:

Find interesting and beautiful places that I would like in New York

Please select sentences that you would like to be in your summary. This will act as an *example* summary.

Western New York is considered part of the Great Lakes region and borders on Lake Ontario, Lake Erie, and Niagara Falls. New York has many state parks and two major forest preserves. Several U.S. national sports halls of fame are situated in New York. The National Baseball Hall of Fame and Museum is located in Cooperstown, Otsego County. New York is represented by Chuck Schumer and ...

Figure 1: Query-based vs. Example-based interface.

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 composing (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 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.

Evaluating an example-based summarization system is more challenging than query-based systems. For each intent, the dataset should include a few different summaries, all produced by the same user. 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 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.

We use SubSume to 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., 2017) and X-Sum (Narayan et al., 2018) datasets, both of which contain single-sentence summaries of news articles.

	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.

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 exactly one associated summary that corresponds 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.

Fremann and Klementiev (2019), in the context of “aspect-based” summarization, provide a dataset having multiple topic-focused summaries 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 vary across individuals.

3 Dataset Description

We now describe our data collection process and design choices, and analyze statistical properties of the dataset. The dataset will be publicly available.

Intents. We devised 10 intents with different degrees of subjectiveness, ranging from mostly objective to mostly subjective, as shown in Figure 2.

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 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 to the summary, and remove a sentence from the summary. A detailed discussion on the interface is in the Appendix.

Task. Each MTurk task (HIT) required a worker to extract sentences from 8 documents to best summarize them according to a given intent, resulting in 8 $(document, intent, summary)$ triplets. To generate unique HITs, we partitioned the set of 48 documents into 6 disjoint sets, each containing 8 documents. We then paired each of the 6 sets with each of the 10 intents, resulting in 60 unique HITs. We repeated the above procedure 5 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 8 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.

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 (see Appendix). For example, for the intent "places you'd like to visit", the annotator flagged a summary as low-quality as it did not contain any location, but arbitrarily chosen sentences. Additionally, we asked each mTurker how they interpreted the task (see Appendix) to verify if their task understanding was correct, and excluded summaries in case it was not.

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 10 intents.

Data Format. We provide SUBSUME in a format to support both query-based and example-driven approaches. Each completed HIT gives us the following information and contributes to 8 data points in SUBSUME: (1) the intent text (one of I1–I10 in Figure 2), (2) one summary for each of the 8 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 in the Appendix.

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 (Figure 2) aligns well with this subjectiveness score (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).

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.

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 8 documents. In the EX setting, we use summaries of 5 documents, chosen at random from the 8 summaries, as example summaries to *learn* the user's intent, and evaluate on the remaining 3 documents. In the QB setting, the baselines summarize the documents using only the query (intent text), and we evaluate on the same set of 3 documents as in the example-driven setting. We repeat this over 10 different splits of the 8 document-summary pairs, and average out results across all splits, and over all data points. We report F_1 scores of the ROUGE-1, ROUGE-2, and ROUGE-L metrics (Lin, 2004) for all the baselines.

4.1 Baselines

We benchmark the following baselines (see Appendix for description and implementation details):

Keyword first extracts keywords from the example summaries or query text, followed by filtering out of sentences with less than t_k keywords. Lastly, 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 distance to the average embedding of the example summaries (query text) and computes a summary using top-k high-scoring sentences in the document.

Pegasus is a state-of-the-art abstractive summarization model (Zhang et al., 2020) based on transformer (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 the CNN-DailyMail dataset.

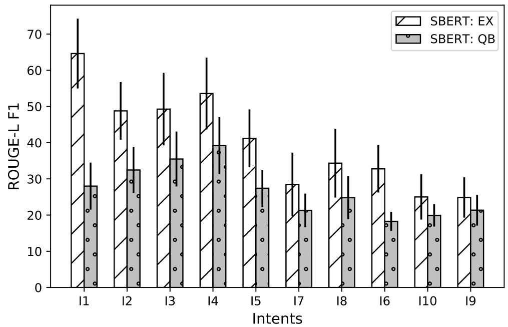


Figure 3: ROUGE-L F_1 for sBert-EX and sBert-QB for each intent. From left to right, intents are ordered in increasing order of their subjectiveness score shown in Table 1. The Pearson's correlation between the subjectiveness score and the F_1 score for sBert-EX and sBert-QB is -0.97 and -0.77 respectively.

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

4.2 Results

Table 2 shows the performance of each baseline averaged over all $(user, intent)$ pairs. 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 can provide examples, example-driven approaches should be preferred 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. As we go from intents with low subjectiveness scores to intents with high subjectiveness scores, performance of sBert decreases for both EX and QB, but sBert-EX consistently outperforms sBert-QB. This shows how the summarization task becomes challenging with increase in subjectiveness of the intents.

These results 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.

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. 2009. Bayesian query-focused summarization. *arXiv preprint arXiv:0907.1814*.

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 R Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. *arXiv preprint arXiv:1906.01749*.

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, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2125–2131. Association for Computational Linguistics.

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, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*.

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, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

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. *arXiv preprint arXiv:1808.08745*.

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*. Association for Computational Linguistics.

Alexander M Rush, SEAS Harvard, Sumit Chopra, and Jason Weston. 2017. A neural attention model for sentence summarization. In *ACLWeb. Proceedings of the 2015 conference on empirical methods in natural language processing*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

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.

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. *CoRR*, abs/1909.01716.

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.