STRUM: Extractive Aspect-Based Contrastive Summarization

Beliz Gunel
Google
Mountain View, USA
bgunel@google.com

Sandeep Tata
Google
Mountain View, USA
tata@google.com

Marc Najork
Google
Mountain View, USA
najork@google.com

ABSTRACT

Comparative decisions, such as picking between two cars or deciding between two hiking trails, require the users to visit multiple webpages and contrast the choices along relevant aspects. Given the impressive capabilities of pre-trained large language models [4, 11], we ask whether they can help automate such analysis. We refer to this task as *extractive aspect-based contrastive summarization* which involves constructing a structured summary that compares the choices along relevant aspects. In this paper, we propose a novel method called STRUM for this task that can generalize across domains *without* requiring any human-written summaries or fixed aspect list as supervision. Given a set of relevant input webpages, STRUM solves this problem using two pre-trained T5-based [11] large language models: first one fine-tuned for aspect and value extraction [14], and second one fine-tuned for natural language inference [13]. We showcase the abilities of our method across different domains, identify shortcomings, and discuss questions that we believe will be critical in this new line of research.

CCS CONCEPTS

• **Computing methodologies → Information extraction**; • **Information systems → Web crawling**.

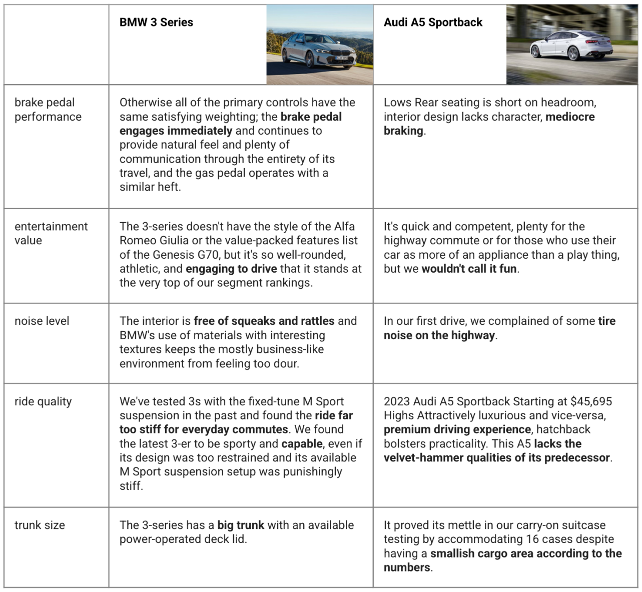
KEYWORDS

contrastive summarization, aspect extraction, entailment models

ACM Reference Format:
Beliz Gunel, Sandeep Tata, and Marc Najork. 2023. STRUM: Extractive Aspect-Based Contrastive Summarization. In *Companion Proceedings of the ACM Web Conference 2023 (WWW '23 Companion), April 30-May 4, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3543873.3587304

1 INTRODUCTION

Many decision tasks require carefully evaluating a small set of choices where there is no single obvious right answer: *"Which espresso grinder should I get?"*, *"Should I go to Stinson Beach or Muir Woods for hiking?"*, *"Could you help me pick between BMW 3 Series and Audi A5 Sportback?"*. For these types of decision tasks, users often need to visit multiple webpages for entities of interest and search for differing relevant aspects between entities to be able to make an informed and systematic decision.



	BMW 3 Series	Audi A5 Sportback
brake pedal performance	Otherwise all of the primary controls have the same satisfying weighting; the **brake pedal engages immediately** and continues to provide natural feel and plenty of communication through the entirety of its travel, and the gas pedal operates with a similar heft.	Lows Rear seating is short on headroom, interior design lacks character, **mediocre braking**.
entertainment value	The 3-series doesn't have the style of the Alfa Romeo Giulia or the value-packed features list of the Genesis G70, but it's so well-rounded, athletic, and **engaging to drive** that it stands at the very top of our segment rankings.	It's quick and competent, plenty for the highway commute or for those who use their car as more of an appliance than a play thing, but we **wouldn't call it fun**.
noise level	The interior is **free of squeaks and rattles** and BMW's use of materials with interesting textures keeps the mostly business-like environment from feeling too dour.	In our first drive, we complained of some **tire noise on the highway**.
ride quality	We've tested 3s with the fixed-tune M Sport suspension in the past and found the **ride far too stiff for everyday commutes**. We found the latest 3-er to be sporty and **capable**, even if its design was too restrained and its available M Sport suspension setup was punishingly stiff.	2023 Audi A5 Sportback Starting at $45,695 Highs Attractively luxurious and vice-versa, **premium driving experience**, hatchback bolsters practicality. This A5 **lacks the velvet-hammer qualities of its predecessor**.
trunk size	The 3-series has a **big trunk** with an available power-operated deck lid.	It proved its mettle in our carry-on suitcase testing by accommodating 16 cases despite having a **smallish cargo area according to the numbers**.

Figure 1: STRUM output summaries comparing BMW 3 Series and Audi A5 Sportback.

We are inspired by the recent rapid success of pre-trained large language models in reasoning tasks for natural language processing [4, 11], and we aim to design our summarization method STRUM to support users in their decision journeys. In particular, we are interested in providing them with a structured output summary where the entities are contrasted per different aspects (say, *brake pedal performance*, *entertainment value*, *noise level*, *ride quality*, and *trunk size* for cars) that are relevant to the domain of interest. Figure 1 shows an example summary comparing two cars generated by STRUM, the technique we describe in the rest of this paper in which we focus on editorial reviews as input text.

We note that existing summarization methods summarize features of a single entity rather than considering multiple entities at the same time [3]. Also, works that produce summaries per each different aspect often have a list of pre-defined fixed aspects to summarize for [1–3], which makes generalizing to different sets of aspects difficult at inference time. Most relevant to our work, Iso et al. [7] propose a contrastive summarization method CoCoSUM along with the publicly available contrastive summarization dataset CoCoTRIP based on hotel reviews. However, they focus on general summaries instead of aspect-based summaries which makes the comparison between two entities less fine-grained, potentially leading to less systematic decisions for users. Also, CoCoSUM is an

WWW '23 Companion, April 30-May 4, 2023, Austin, TX, USA

ACM ISBN 978-1-4503-9419-2/23/04.
https://doi.org/10.1145/3543873.3587304

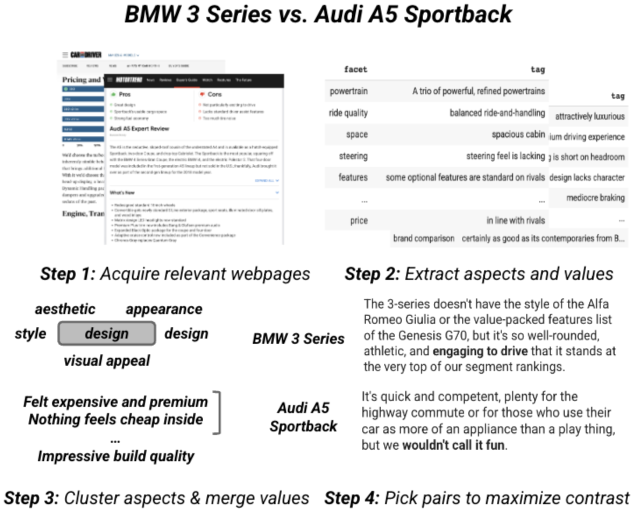
abstractive summarization method that is susceptible to hallucination of incorrect information [1]. We prioritize factual correctness, hence we design our summarization method to be extractive. We believe that more contrast across aspects is helpful for users to reach a decision between different options. Unlike CoCoSUM that considers token-level overlap while defining contrast, we consider *semantic contrast* as two pieces of text can express similar meaning with different tokens or different meaning with the same tokens.

In this paper, we tackle *extractive aspect-based contrastive summarization*, which translates to constructing a contrastive summary comparing relevant aspects of two entities with summary sentences directly from the source text. We propose a novel method called STRUM for this task that involves four steps: (1) acquiring relevant webpages for entities, (2) extracting aspects and corresponding values, (3) merging extracted aspects and their corresponding values to achieve a balance between aspects that are too fine-grained or too coarse, and (4) picking source sentences to include in the summary that maximize the contrast for each aspect. In three of these steps we use a pre-trained T5-based [11] large language model fine-tuned for the task: aspect extraction for step 2; and natural language inference (entailment) for steps 3 and 4. We explain our design decisions for STRUM in Section 3 including extractive vs. abstractive summarization, aspect extraction vs. general summaries, clustering aspects and values, and using an entailment model as a proxy for similarity/contrast. In Section 4, we present examples to showcase STRUM in different domains, and discuss both the strengths and the weaknesses of our method. In Section 5, we discuss directions that we believe will be critical for this line of research.

2 RELATED WORK

Existing summarization techniques primarily focus on summarizing popular features of a single entity instead of comparing two entities. Among representative works in this line of research, Angelidis et al. [3] introduce a quantized transformer for aspect-based extractive opinion summarization using a discretization bottleneck of vector-quantized variational autoencoders [12]. Amplayo et al. [2] propose to construct a synthetic training dataset through using a random review as the pseudo-summary among a set of reviews along with three different types of aspect controllers with varying levels of granularity to fine-tune a pre-trained language model on for aspect-controllable opinion summarization. Ahuja et al. [1] construct a new extractive aspect-based news summarization dataset for earthquakes and fraud reports from CNN/DailyMail along with a method that focuses on generalization to new aspect types. Related to contrastive summarization, Lerman and McDonald [8] propose a method based on statistical language models which primarily considers sentiments of opinions, and they evaluate on consumer reviews. Finally, Iso et al. [7] propose a neural contrastive summarization approach that they evaluate on a set of short hotel reviews. Their core contribution is a method called *co-decoding* that contrasts token probability distributions for contrastive summaries and aggregates them for common summaries. Note that in comparison to our aim, (1) this approach is abstractive instead of extractive, (2) generates general summaries instead of aspect-based summaries, and (3) focuses on token-level contrast instead of semantic contrast that we get from using an entailment model.

3 STRUM DESCRIPTION



The 3-series doesn't have the style of the Alfa Romeo Giulia or the value-packed features list of the Genesis G70, but it's so well-rounded, athletic, and **engaging to drive** that it stands at the very top of our segment rankings.

It's quick and competent, plenty for the highway commute or for those who use their car as more of an appliance than a play thing, but we **wouldn't call it fun**.

Figure 2: STRUM overview.

In this section, we describe our method STRUM and outline the main design decisions. We include a diagram of our overall STRUM approach in Figure 2. As input, we specify (1) two entities, say, *BMW 3 Series* and *Audi A5 Sportback* for the decision of picking a car to buy; and (2) corresponding lists of webpages that are the top results in web search. Note that the user can specify the number of webpages to include as input, and by default we pick the first three webpages returned by a search engine when provided with the query “*entity* + review”. We extract the essential sentences in the input webpages for both entities and tile them into chunks that contain a few sentences each. In all our presented results, we use a length of 256 characters while dividing extracted sentences into chunks, as aspect discovery model performs considerably better on shorter input text. First, we run an aspect extraction model [14] that is based on a pre-trained large language model fine-tuned on shopping-related data for high-precision attribute understanding on text for both entities. For example, from the input text “*The larger the screen, the heavier your cell phone will be.*”, aspect extraction model would extract *screen size* and *weight* as aspects and *larger* and *heavier* as corresponding values. This model does not require a pre-specified fixed aspect list. Note that the terms facet, attribute, and aspect are used interchangeably throughout the paper. The level of fine-grainedness across aspects can differ across webpages, domains, and entities. Hence, we merge aspects through an agglomerative hierarchical clustering approach [10] based on a similarity threshold that is set as a hyperparameter. We measure similarity between facets using a pre-trained natural language inference entailment model that we describe further in our design decisions below. After clustering the discovered facets for each entity, we merge values of these facets again based on their entailment model similarities using a threshold that is similarly set as a hyperparameter. Finally, for each shared aspect, we pick source sentences for each aspect that maximizes the contrast using an entailment model. User can specify the maximum number of sentences per each aspect and maximum number of total summary sentences. We discuss our choice of using an entailment model as a proxy for similarity (and contrast) below as part of our design decisions. Also, we further describe how each stage is designed. In our final output structured summaries, each row corresponds to a single facet with a piece-of-text along that facet for both of the entities.

3.1 Design Decisions

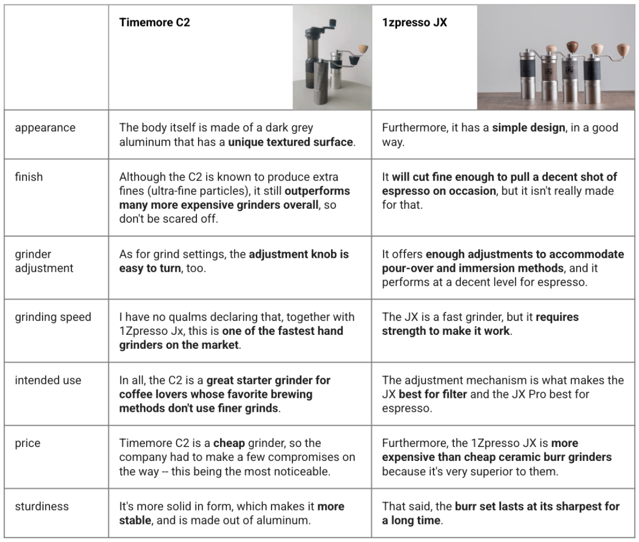
Extractive Summarization: Extractive summarization is defined as the class of methods that contain sentences directly from the source text, while abstractive methods paraphrase information from the source. Abstractive summarization methods are more human-like, but they can hallucinate information that is not factually consistent with information in the source documents, and they are hard to evaluate due to variance and subjectivity of human raters during ground truth summary collection [1]. Note that although recent pre-trained large language model based technologies such as T5 [11] or PaLM [4] possess impressive capabilities in presenting information, they do not have the ability to fully ground (i.e. show links for source webpages) the information they show. On the other hand, extractive methods are easy to ground, and are easy to evaluate using automated word or n-gram overlap metrics such as ROUGE [9]. In STRUM, we prioritize producing factually correct summaries over being able to generate novel text, hence we design STRUM to be fully extractive.

Aspect Extraction: Existing opinion summarization methods either provide general summaries that include all the aspects without paying attention to each one specifically [7] or summarize based on a fixed set of pre-defined aspects [3]. In contrast, STRUM extracts set of aspects that are most relevant to the particular decision and compares both entities across each aspect so that user can make an informed and systematic decision. Note that having a fixed set of pre-defined aspects makes (say, cleanliness, price, location, for the hotels domains) it much harder to generalize to a different domains (say campsites) and personalize to user preferences. For further details on the aspect extractor we use, refer to Vilnis et al. [14].

Clustering of Aspects and Values: Aspect extraction can be very fine-grained, hence it can be hard to find shared aspects across entities without some form of post-processing. As an example, consider a case where the aspect extractor model identifies *benefits*, *card benefits*, *card perks*, and *credit card rewards* as distinct aspects. We propose to group these aspects into a general one and compare two entities based on that general one. Specifically, we use a hierarchical agglomerative clustering approach [10] while merging different facets where we use *symmetric facet similarity* as the distance function. We define symmetric facet similarity as (**ent**(facet1 sentence, facet2 sentence)+**ent**(facet2 sentence, facet1 sentence))/2 where **ent** is the entailment model and the sentences are constructed using "$\{entity\}\{facet\}is\{val\}$." template. For the entailment model, we utilize a pre-trained T5-11B encoder-decoder language model [11] fine-tuned on natural language inference (NLI) task that involves automatically determining whether the meaning of the hypothesis can be inferred from the premise [5]. Note that there has been previous work that utilized entailment models to measure semantic similarity between two pieces of text for factual consistency applications [6]. In particular, we use a pre-trained sentence-pair NLI model that was fine-tuned on several well-established NLI datasets to increase the model’s robustness to longer form and out-of-distribution inputs [13]. To compare aspects with an NLI model, we construct both the premise and the hypothesis using the template "$\{entity\}\{facet\}is\{val\}$.", where we use both a common entity string and a common value string. Once facets are clustered, we merge values for each facet using *value similarity* defined as **max**(**ent**(value1 sentence, value2 sentence), **ent**(value2 sentence, value1 sentence)) where we construct values sentences similarly as "$\{entity\}\{facet\}is\{val\}$.". Again, we use both a common entity string and common facet string. We set the similarity threshold as a hyperparameter that we tune based on the domain while both clustering facets and merging values.

Picking Contrastive Sentences: For each extracted shared aspect between two entities, we pick the source sentences that provide the least redundancy (most contrast) using the entailment model described above following the formula of **max**(**ent**(pseudo summary sentence1, pseudo summary sentence2), **ent**(pseudo summary sentence2, pseudo summary sentence1)). To construct the pseudo summary sentences, we use a "$\{entity\}\{facet\}is\{val\}$." template instead of the actual source sentences to simplify the complex sentence structure and instead help the entailment model to focus on the entity and the facet, where we use a common entity string.

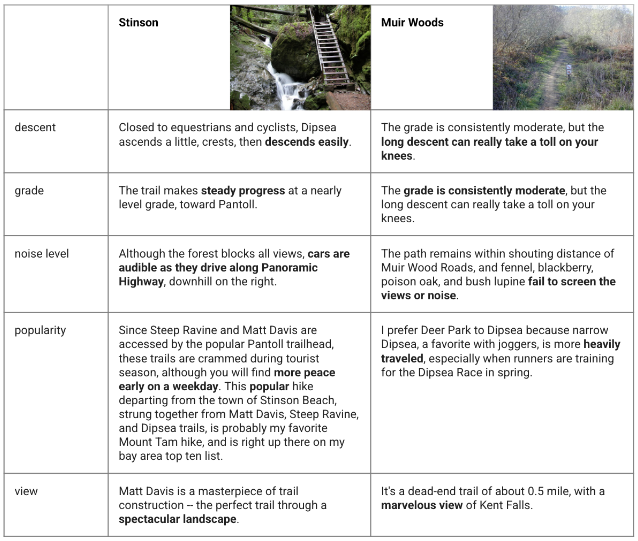
4 STRUM DEMO



	Timemore C2	1zpresso JX
appearance	The body itself is made of a dark grey aluminum that has a **unique textured surface**.	Furthermore, it has a **simple design**, in a good way.
finish	Although the C2 is known to produce extra fines (ultra-fine particles), it still **outperforms many more expensive grinders overall**, so don't be scared off.	It **will cut fine enough to pull a decent shot of espresso on occasion**, but it isn't really made for that.
grinder adjustment	As for grind settings, the **adjustment knob is easy to turn**, too.	It offers **enough adjustments to accommodate pour-over and immersion methods**, and it performs at a decent level for espresso.
grinding speed	I have no qualms declaring that, together with 1Zpresso Jx, this is **one of the fastest hand grinders on the market**.	The JX is a fast grinder, but it **requires strength to make it work**.
intended use	In all, the C2 is a **great starter grinder for coffee lovers whose favorite brewing methods don't use finer grinds**.	The adjustment mechanism is what makes the JX **best for filter** and the JX Pro best for espresso.
price	Timemore C2 is a **cheap** grinder, so the company had to make a few compromises on the way -- this being the most noticeable.	Furthermore, the 1Zpresso JX is **more expensive than cheap ceramic burr grinders** because it's very superior to them.
sturdiness	It's more solid in form, which makes it **more stable**, and is made out of aluminum.	That said, the **burr set lasts at its sharpest for a long time**.

Figure 3: STRUM output summaries comparing Timemore C2 and 1zpresso JX coffee grinders.

We provide output summaries for three examples from different domains in order to showcase the abilities of STRUM: we compare BMW 3 Series and Audi A5 Sportback (Figure 1); Timemore C2 and 1zpresso JX coffee grinders (Figure 3); and Stinson and Muir Woods hikes (Figure 4). Each row in the output summary corresponds to a discovered aspect that is shared across both entities. An image for each entity is also extracted from the input webpages that are the top search results. We bold the text that corresponds to the value



	Stinson	Muir Woods
descent	Closed to equestrians and cyclists, Dipsea ascends a little, crests, then **descends easily**.	The grade is consistently moderate, but the **long descent can really take a toll on your knees**.
grade	The trail makes **steady progress** at a nearly level grade, toward Pantoll.	The **grade is consistently moderate**, but the long descent can really take a toll on your knees.
noise level	Although the forest blocks all views, **cars are audible as they drive along Panoramic Highway**, downhill on the right.	The path remains within shouting distance of Muir Wood Roads, and fennel, blackberry, poison oak, and bush lupine **fail to screen the views or noise**.
popularity	Since Steep Ravine and Matt Davis are accessed by the popular Pantoll trailhead, these trails are crammed during tourist season, although you will find **more peace early on a weekday**. This **popular** hike departing from the town of Stinson Beach, strung together from Matt Davis, Steep Ravine, and Dipsea trails, is probably my favorite Mount Tam hike, and is right up there on my bay area top ten list.	I prefer Deer Park to Dipsea because narrow Dipsea, a favorite with joggers, is more **heavily traveled**, especially when runners are training for the Dipsea Race in spring.
view	Matt Davis is a masterpiece of trail construction -- the perfect trail through a **spectacular landscape**.	It's a dead-end trail of about 0.5 mile, with a **marvelous view** of Kent Falls.

Figure 4: STRUM output summaries comparing Stinson and Muir Woods hikes.

of the extracted aspect for each entity. Recall that we show the sentences extracted from the source text that provide the highest contrast between entities for that particular aspect. The user has the ability to control the maximum number of sentences shown per aspect and total maximum number of sentences in the output summary. We would like to point out that although the aspect extractor was fine-tuned on shopping data, STRUM can generalize to non-shopping domains such as hikes. Note that STRUM is able to provide useful contrast between entities for shared facets such as pointing out *brake pedal engaging immediately* for BMW 3 Series while Audi A5 Sportback has *mediocre braking* for brake pedal performance (Figure 1). It currently does not provide information on non-shared facets (e.g., battery capacity vs. fuel tank size) which is left for future work. Also, STRUM currently does not provide context on how prevalent these opinions are within the reviews in input webpages – sharing a similar challenge with existing summarization methods.

5 DISCUSSION ON NEW RESEARCH DIRECTIONS

First and foremost, there needs to a public benchmark for the extractive aspect-based contrastive summarization task along with established automated evaluation metrics specific to the task. To the best of our knowledge, the only public contrastive summarization benchmark available is the CoCoTRIP dataset [7] that provides human-written ground truth general abstractive summaries for 48 entity pairs drawn from the hotels domain; and the only task-specific automated evaluation metric introduced in that work is the *distinctiveness score* based on token overlap. Second, our solution can be improved for the individual users if we take personalization to their preferences into account. This would entail robustly understanding the user intent to map to the relevant aspects in order to include in the contrastive summary. Third, inconsistency within the input where reviews do not agree with each other makes this task harder. There can be multiple ways to tackle this including reporting the majority opinion and providing the distribution of opinions. Although STRUM initially focused on editorial reviews, we would like to extend to user-submitted reviews. Fourth, most machine learning based solutions including ours perform worse in the case of longer-form input text (i.e. aspect discovery works significantly better on shorter text in our case which is a critical component in our solution), hence there should be more investigations into how to effectively extend solutions to longer-form text. Finally, integration of multi-modal inputs such as images, video, and speech into the contrastive summarization would surely help the users with their decision journey.

6 CONCLUSION

In this paper, we tackle extractive aspect-based contrastive summarization to help users for decision tasks such as picking between two cars. We propose a novel method STRUM that can generalize across domains with no human supervision through human-written summaries or an aspect list. We provide compelling extractive contrastive summaries for entity pairs in each domain that do not contain hallucinations and focus on semantic contrast between sentences. Finally, we discuss new directions that we believe will be critical for this line of research.

REFERENCES

[1] Ojas Ahuja et al. 2022. ASPECTNEWS: Aspect-Oriented Summarization of News Documents. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 6494–6506. https://doi.org/10.18653/v1/2022.acl-long.449

[2] Reinald Kim Amplayo et al. 2021. Aspect-Controllable Opinion Summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 6578–6593. https://doi.org/10.18653/v1/2021.emnlp-main.528

[3] Stefanos Angelidis et al. 2020. Extractive Opinion Summarization in Quantized Transformer Spaces. *Transactions of the Association for Computational Linguistics* 9 (2020), 277–293. https://doi.org/10.1162/tacl_a_00366

[4] Aakanksha Chowdhery et al. 2022. PaLM: Scaling Language Modeling with Pathways. *ArXiv* abs/2204.02311 (2022).

[5] Ido Dagan et al. 2007. The PASCAL Recognising Textual Entailment Challenge. In *Machine Learning Challenges Workshop*. https://doi.org/10.1007/11736790_9

[6] Or Honovich et al. 2021. Q^2: Evaluating Factual Consistency in Knowledge-Grounded Dialogues via Question Generation and Question Answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 7856–7870. https://aclanthology.org/2021.emnlp-main.619/

[7] Hayate Iso et al. 2022. Comparative Opinion Summarization via Collaborative Decoding. In *Findings of the Association for Computational Linguistics: ACL 2022*. 3307–3324. https://aclanthology.org/2022.findings-acl.261

[8] Kevin Lerman and Ryan McDonald. 2009. Contrastive Summarization: An Experiment with Consumer Reviews. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*. Association for Computational Linguistics. https://aclanthology.org/N09-2029

[9] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, 74–81. https://aclanthology.org/W04-1013

[10] Frank Nielsen. 2016. *Hierarchical Clustering*. 195–211. https://doi.org/10.1007/978-3-319-21903-5_8

[11] Colin Raffel et al. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67. http://jmlr.org/papers/v21/20-074.html

[12] Aurko Roy et al. 2018. Theory and Experiments on Vector Quantized Autoencoders. *ArXiv* abs/1805.11063 (2018).

[13] Tal Schuster et al. 2022. Stretching Sentence-pair NLI Models to Reason over Long Documents and Clusters. *Findings the Conference on Empirical Methods in Natural Language Processing* abs/2204.07447 (2022).

[14] Luke Vilnis et al. 2022. ImPaKT: A Dataset for Open-Schema Knowledge Base Construction. *ArXiv* abs/2212.10770 (2022).