On the Role of Summary Content Units in Text Summarization Evaluation

Anonymous EMNLP submission

Abstract

At the heart of the pyramid evaluation method for text summarization lie human written summary content units (SCUs). These SCUs are concise sentences that decompose a summary into small facts. Such SCUs can be used to judge the quality of a candidate summary, possibly partially automated via natural language inference (NLI) systems. Interestingly, with the aim to fully automate the pyramid evaluation, Zhang and Bansal (2021) show that SCUs can be approximated from parsed semantic role triplets (STUs). However, several questions currently lack answers, in particular i) Are there other ways of approximating SCUs that can offer advantages? ii) Under which conditions are SCUs (or their approximations) offering the most value? In this work, we examine two novel strategies to approximate SCUs: generating SCU approximations from AMR meaning representations (SMUs) and from large language generation models (SGUs), respectively. We find that while STUs and SMUs are competitive, the best approximation quality is achieved by SGUs. We also show through a simple sentence-decomposition baseline (SSUs) that SCUs (and their approximations) offer the most value when ranking short summaries, but may not help as much when ranking systems or longer summaries.

1 Introduction

Judging the quality of a summary is a challenging task. Besides being short and faithful to its source document, a summary should particularly excel in *relevance*, that is, a summary should select only the most relevant or salient facts from a source document. An attractive method for assessing such notion of relevance is the *Pyramid*-method (Nenkova and Passonneau, 2004) that is based on so-called *Summary Content Units* (SCUs) which decompose a reference summary into concise human-written English sentences. With SCUs available from one or different reference summaries, we can then more objectively assess the degree to which a candidate summary contains the relevant information. With the aim to fully automate the Pyramid method, Zhang and Bansal (2021) suggest that the required human effort can be partially and even fully alleviated, by i) automatically generating SCUs and ii) validating the relevance of a summary with an NLI system that checks how many SCUs are entailed by a candidate summary.

Since strong NLI systems are available off-the-shelf and are known to be useful in NLG evaluation[1], clearly the generation of SCUs is the most challenging and least-understood part of an automated pyramid. Indeed, while Zhang and Bansal (2021) show that SCUs can be approximated by phrasing semantic role parsed triplets, we lack availability and understanding of possible alternatives as well as their potential impact on downstream-task summary evaluation in different scenarios.

In this work, we proposed two novel approaches to approximate SCUs: SMUs that are based on abstract meaning representation (AMR) and SGUs that leverage SoTA large language models (LLMs). We carry out experiments to systematically evaluate the intrinsic quality of SCUs and their approximations. On the downstream task evaluation, we find that although SCUs remain the most effective metric to rank different systems or summaries across three meta-evaluation datasets, surprisingly, an efficient sentence splitting baseline already yields competitive results when compared to SCUs. In fact, the sentence splitting baseline outperforms the best SCU approximation method on a few datasets when ranking systems or long summaries.

In summary, our work provides important insights into the application of automation of the pyramid method in different scenarios for evaluating summaries. We make the code publicly available at [URL Upon Acceptance].

[1] E.g., see Chen and Eger (2022), or Steen et al. (2023).

2 Related work

Evaluating the quality of a summary is a challenging task. Over the past two decades, researchers have proposed a wide range of human-in-the-loop or automatic metrics to assess summaries in different dimensions, including linguistic quality, coherence, faithfulness, and content quality. For more in-depth surveys on this topic, please refer to Howcroft et al. (2020) and Gehrmann et al. (2022).

In this work, we focus on evaluating the content quality of a summary that assesses whether the summary effectively captures the salient information of interest from the input document(s). In the reference-based metrics, content quality is assessed by comparing system-generated summaries to human-written reference summaries. The pyramid method (Nenkova and Passonneau, 2004) is regarded as a reliable and objective approach to evaluating the content quality of a summary. Below we briefly describe the pyramid method and highlight some previous efforts to automate this method.

Pyramid Method. The original pyramid method (Nenkova and Passonneau, 2004) comprises two steps: SCUs creation and system evaluation. In the first step, human annotators exhaustively identify Summary Content Units (SCUs) from the reference summaries. Each SCU is a concise sentence and describes a single fact. The weight of an SCU is determined by the number of references in which it occurs. In the second step, the presence of each SCU in a system summary is manually checked. The system summary's pyramid score is calculated as the normalized sum of the weights of the SCUs that are present. Later, Shapira et al. (2019) introduce a revised version of the original pyramid method where they eliminate the merging and weighting of SCUs, thereby enabling SCUs with the same meaning to coexist.

Automation of the Pyramid Method. Given the high cost and the expertise required for implementing the pyramid method, in recent years there are a few attempts to automate this approach. Peyrard et al. (2017) propose an automatically learned metric to directly predict human pyramid scores based on a set of features. Zhang and Bansal (2021) propose a system called $Lite^3Pyramid$ that uses a semantic role labeller to extract semantic triplet units (STUs) to approximate SCUs. They further use a trained natural language inference (NLI) model to replace the manual work of assessing SCUs' presence in system summaries. In our work, we explore two new methods to approximate SCUs. We further investigate the effectiveness of the automated pyramid method in different scenarios.

3 SCU approximation I: SMU from AMR

Abstract Meaning Representation (AMR) (Banarescu et al., 2013) is a widely-used semantic formalism employed to encode the meaning of natural language text in the form of rooted, directed, edge-labeled, and leaf-labeled graphs. The AMR graph structure facilitates machine-readable explicit representations of textual meaning.

Motivated by Zhang and Bansal (2021)'s observation that STUs based on semantic roles cannot well present single facts in long reference summary sentences that contain a lot of modifiers, adverbial phrases, or complements, we hypothesize that AMR has the potential to capture such factual information more effectively. This is because, in addition to capturing semantic roles, AMR models finer nuances of semantics, including negations, inverse semantic relations, and coreference.

To generate semantic meaning units (SMUs) from a reference summary, we first leverage a pre-trained Text2AMR model[2] to represent each sentence in the summary as an AMR graph. We then design a few heuristics to split each AMR graph into several sub-graphs and apply an AMR2Text model[3] on each sub-graph to generate SMUs. Please refer to Appendix A.1 for more details on splitting an AMR graph into multiple sub-graphs.

4 SCU approximation II: SGU from LLM

Recently, it became widely known that pre-trained large language models (LLMs) are able to generate high-quality output according to prompts given by humans, optionally exploiting shown examples through in-context learning (Brown et al., 2020). Therefore, we try to approximate SCUs using GPT models from OpenAI, calling the resulting units as Semantic GPT Units (SGUs). Specifically, we use GPT-3.5-Turbo which is built on InstructGPT (Ouyang et al., 2022) and GPT-4 (OpenAI, 2023) to generate SGUs (SGUs_3.5 and SGUs_4) for each reference summary using the same prompt and a

[2] *parse_xfm_bart_large* (https://github.com/bjascob/amrlib-models)

[3] generate_t5wtense (https://github.com/bjascob/amrlib-models)

one-shot example. Please refer to Appendix A.2 for more details.

5 Experiments

5.1 Dataset and NLI models

Data. We run the experiments on four existing English meta-evaluation datasets: (1) TAC08 (NIST, 2008), (2) TAC09 (NIST, 2009), (3) REALSumm (Bhandari et al., 2020) and (4) PyrXSum (Zhang and Bansal, 2021) and evaluated the results on the last three datasets, using TAC08 for development purposes. Each dataset contains one or multiple reference summaries, the corresponding human-written SCUs, the generated summaries from different systems, and the human evaluation result for each summary/system based on the pyramid method. Table 1 shows some statistics of the reference summaries across different datasets. In general, PyrXSum contains short and abstractive summaries, while RealSumm and TAC09 contain long and extractive summaries. More details on the datasets can be found in appendix A.4.

NLI Models. We use the NLI model from Zhang and Bansal (2021) that was fine-tuned on TAC08's SCU presence gold annotations based on a NLI model from Nie et al. (2020).

5.2 Baselines

STUs are the semantic role triples based on semantic role labelling (Zhang and Bansal, 2021).

Sentence splitting is a baseline that may shed light on the overall usefulness of SCUs in summary evaluation. We split every reference summary into sentences and treat them as SCU approximations.

N-grams consist of phrases randomly extracted from a reference summary. For each sentence from the summary, we naïvely generate all possible combinations of 3, 4, and 5 consecutive words. We then randomly select a subset from these combinations, which accounts for 5% of the total number of n-grams produced.

5.3 Intrinsic Evaluation

As proposed by Zhang and Bansal (2021), we evaluate approximation quality with an *easiness score*. The score is built by iterating over each SCU-SxU pair and average over the maximum ROUGE-1-F1 score found for each SCU. Naturally this score is recall-biased, and therefore, we also present the score calculated in the reverse direction, to assess whether our approximated SCUs are of high precision.

	RealSumm	PyrXSum	TAC09
Avg. # sent.	4.73	2.02	27.22
Avg. # words	63.71	20.56	386.82
Avg. # words/sent	13.47	10.18	14.21
# ref summary	1	1	4
Avg. # SCUs	10.56	4.78	31.63

Table 1: Statistics of the reference summaries from different datasets.

	RealSumm		PyrXSum		TAC09	
Metrics	R	P	R	P	R	P
sentence split	.54	.67	.41	.54	.50	.54
ngrams	.41	.52	.38	.52	.46	.39
STUs	**.66**	.68	.54	.65	**.61**	.53
SMUs	.56	.58	.53	.58	.52	.48
SGUs_3.5	.58	.67	.58	.63	.36	.48
SGUs_4	.61	**.69**	**.61**	**.66**	.52	**.61**

Table 2: Intrinsic Evaluation Results. R is the recall oriented simulation easiness score by (Zhang and Bansal, 2021), while P is our precision-oriented score that is computed in the reverse direction.

The results are shown in Table 2. We find that best approximation quality for RealSumm is achieved by STUs, while for PyrXSum, SGU_4 performs best. Considering the longer texts of TAC09, STUs excel in recall, while SGU_4 excels in precision.

5.4 Extrinsic Evaluation

Our downstream evaluation consists of two parts: summary quality evaluation at the system and summary levels, respectively. System-level correlation assessment evaluates the ability of the metric to compare different summary systems individually. In contrast, summary-level evaluation determines the metric's ability to compare summaries created by different systems for a common set of documents. Following (Zhang and Bansal, 2021), we use Pearson r and Spearman ρ to evaluate the correlations between metrics with gold human labelling scores. Pearson measures linear correlation and Spearman measures ranking correlation. See more details on how to compute these two types of correlation in appendix A.3.

The results are shown in Table 3. We can observe that SGUs overall offer the most useful SCU approximation, with the exception for TAC09

	System-Level						Summary-Level					
	RealSumm		PyrXSum		TAC09		RealSumm		PyrXSum		TAC09	
Metrics	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
SCUs	.95	.95	.98	.98	.99	.97	.59	.58	.70	.69	.76	.70
SCU Approximations												
- sentence split	.93	**.95**	**.97**	**.97**	.97	.94	.48	.46	.37	.36	**.73**	.66
- ngrams	.90	.92	.94	.82	.96	.92	.36	.35	.38	.38	.65	.61
- STUs	.92	.94	.95	.95	**.98**	.95	.51	.50	.46	.44	**.73**	**.67**
- SMUs	**.94**	.94	.96	.94	**.98**	**.96**	.50	.48	.46	.44	.70	.64
- SGUs_3.5	.93	**.95**	**.97**	.93	.96	.88	.49	.46	.56	.55	.54	.49
- SGUs_4	.92	.94	**.97**	.95	**.98**	**.96**	**.54**	**.52**	**.58**	**.56**	.71	.66

Table 3: Results of different metrics on three datasets. Best numbers among all SCU approximations are bolded.

(summary-level), where STUs remain the best approximation method, slightly outperforiming our simple sentence splitting baseline. However, SGUs still lack the usefulness of true SCUs, which seem to remain the most useful way to evaluate summary quality (if resources permit). Interestingly, however, to discriminate the quality of systems, it is enough to use any approximation, even the sentence split baseline is sufficient to accurately discriminate systems.

5.5 Human Evaluation

For a representative sample of human results of our experiment, three authors evaluated the quality of SCUs, STUs, SMUs and SGUs_4 for 10 reference summaries randomly sampled from REALSumm and PyrXSum. Cohen's κ scores among three annotators range from 0.37 to 0.87. More details about human evaluation can be found in Appendix A.5.

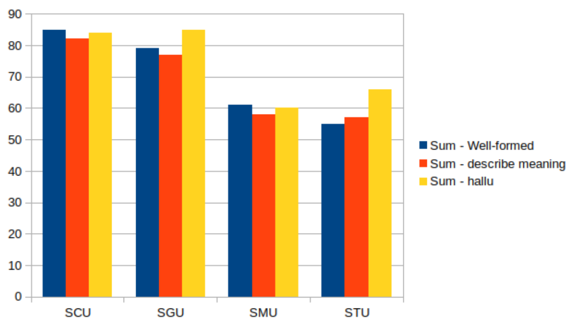

Figure 1: Human evaluation results. Numbers on the y-axis represent the aggregated scores of all three annotators for the 10 examples, with higher scores indicating better performance across all three dimensions.

The findings presented in Figure 1 illustrate that the quality of SMUs is comparable to the STUs. SMUs are a little bit more well-formed but hallucinate more compared to STUs. Furthermore, it's important to note that SGUs are nearly on par with SCUs in terms of overall quality. However, despite their close performance, SGUs exhibit certain shortcomings. They lack a certain degree of Well-formedness, and Descriptiveness. Thus, while SGUs and SCUs might appear similar in performance, a closer inspection reveals a slightly better performance of the SCUs.

5.6 Discussion

In our research, we found out that there are more effective ways of approximating SCUs than only with STUs. There are several aspects worth discussion. Firstly, it appears that the automatic intrinsic evaluation metric, based on ROUGE-1-F1, exhibits a weak correlation with human evaluation. This raises concerns about the effectiveness of using this metric in previous studies to evaluate the quality of SCU approximations. Secondly, it seems that we may not need the costly SCUs and their approximations to compare summarization systems or rank long summaries (TAC09). Surprisingly, a simple sentence splitting baseline already achieves competitive results compared to SCUs. Finally, SCUs and their approximations offer the most value to rank short summaries (PyrXSum and RealSumm).

6 Conclusions

This work primarily focuses on automating the pyramid method. We propose two new methods to approximate SCUs and systematically evaluate the intrinsic quality of SCUs and their approximations. Our experiments on extrinsic evaluation suggests that there might be no need for costly SCUs and their approximations when comparing summarization systems.

Limitations

Limitations

First, we would have liked to achieve better performance with SMUs generated from AMR, since, in theory, AMR graph splitting seems ideal to decompose a textual meaning into parts, and AMR generation systems promise to phrase any such subgraph in natural language. Inspecting all three parts of the pipeline (parsing, splitting, and generating), we find that most issues are likely due to our manually designed splitting strategy. While the rules are simple and their creation has profited from communication with AMR-knowledgeable researchers, the main problem is that there are countless possibilities of how to split an AMR, and the importance of rules may strongly depend on the further graph context. Therefore, we believe it is likely that future work can strongly improve the AMR approach by better learning how to better split meaning representation graphs.

Second, we used an NLI system that was fine-tuned on gold SCUs extracted from the development data (TAC08), since this NLI system was found to work best by Zhang and Bansal (2021). While in principle this does not affect the evaluation of SxUs, which was the focus of this paper, it is not unlikely that by training the NLI system on each SxU type separately, the results of SxUs may further improve and so the results for human SCUs can be considered as slightly optimistic. In general, the interaction of automatic NLI and SCUs in an automated pyramid needs to be better understood. Other recent findings (Chen and Eger, 2022; Steen et al., 2023) suggest that NLI models may play an underestimated role in NLG evaluation. As a check, we repeated evaluation with an NLI system without SCU fine-tuning, and observe significant performance drops across the board, indicating that i) SCU results are likely not too over-optimistic in comparison to SxUs; and ii) the effective adaptation strategy of the NLI system indeed may be the second cornerstone of an accurate automatic pyramid and therefore should be better explored in future work.

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, Pengfei Liu, and Graham Neubig. 2020. Re-evaluating evaluation in text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9347–9359, Online. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Yanran Chen and Steffen Eger. 2022. Menli: Robust evaluation metrics from natural language inference. *arXiv preprint arXiv:2208.07316*.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2022. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NIPS*.

David M. Howcroft, Anya Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A. Hasan, Saad Mahamood, Simon Mille, Emiel van Miltenburg, Sashank Santhanam, and Verena Rieser. 2020. Twenty years of confusion in human evaluation: NLG needs evaluation sheets and standardised definitions. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 169–182, Dublin, Ireland. Association for Computational Linguistics.

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, Brussels, Belgium. Association for Computational Linguistics.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 145–152, Boston, Massachusetts, USA. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

NIST. 2008. https://tac.nist.gov/publications/2008/papers.html.

NIST. 2009. https://tac.nist.gov/publications/2009/papers.html.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Maxime Peyrard, Teresa Botschen, and Iryna Gurevych. 2017. Learning to score system summaries for better content selection evaluation. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 74–84, Copenhagen, Denmark. Association for Computational Linguistics.

Ori Shapira, David Gabay, Yang Gao, Hadar Ronen, Ramakanth Pasunuru, Mohit Bansal, Yael Amsterdamer, and Ido Dagan. 2019. Crowdsourcing lightweight pyramids for manual summary evaluation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 682–687, Minneapolis, Minnesota. Association for Computational Linguistics.

Julius Steen, Juri Opitz, Anette Frank, and Katja Markert. 2023. With a little push, nli models can robustly and efficiently predict faithfulness. *arXiv preprint arXiv:2305.16819*.

Shiyue Zhang and Mohit Bansal. 2021. Finding a balanced degree of automation for summary evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6617–6632, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Appendix

A.1 Obtaining AMR Sub-graphs

Given an AMR graph, we first extract all predicates to discern their semantic meaning as we view them to form the core of a sentence's meaning. Subsequently, the argument connections within the predicates were examined. If a predicate is connected to at least one core role (CR), indicated by $\texttt{ARG}_n$ edge label, we extract a sub-graph for every CR of this predicate containing the CR and the underlying connections. Below we show an example of extracting sub-graphs (Figure 3 and Figure 4) from an AMR graph (Figure 2) and the corresponding SMUs.

Input sentence: Godfrey Elfwick recruited via Twitter to appear on World Have Your Say.

SMUs from sub-graphs: Godfrey Elfwick was recruited. # Godfrey Elfwick will appear on World Have Your Say.

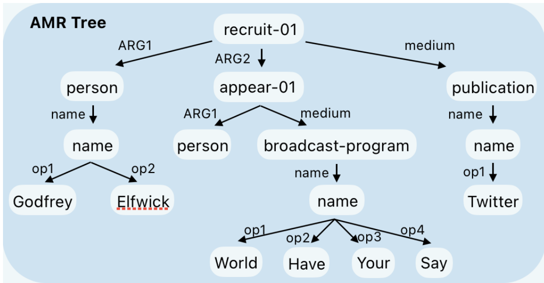

Figure 2: The AMR graph for the sentence "Godfrey Elfwick recruited via Twitter to appear on World Have Your Say"

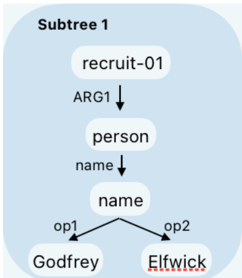

Figure 3: Sub-graph 1

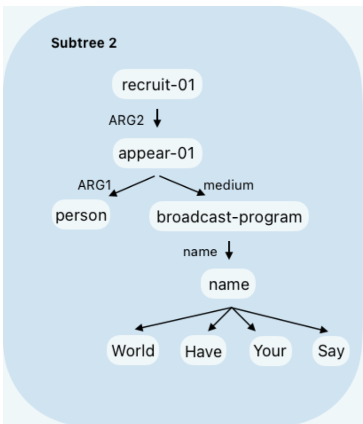

Figure 4: Sub-graph 2

A.2 Obtaining SGUs from the GPT Models

Below we show an example of the prompt we use to obtain SGUs from the GPT-3.5-Turbo and GPT-4 models through OpenAI APIs[4]. We randomly choose a summary and the corresponding human-written SCUs from the TAC08 dataset as the one-shot example. We did not alter the standard parameters except for temperature which we set to 0 to ensure reproducability.

Instruction You split the provided input in small sentences separated by an #. The split sentences represent subsentences of the original sentences.

Example Input Irish PM Ahern said the main goal of the US-brokered Good Friday pact of 1998, a joint Catholic-Protestant administration in Northern Ireland, could be revived only with a complete end of IRA weapons use. The landmark peace deal led to a virtual end of violence in that area. Sinn Fein leader Gerry Adams has appealed to IRA members to end their armed struggle in favor of democratic politics. Hopes are rising in Northern Ireland that the IRA will disarm. British PM Blair and Ahern will chair a review of the Northern Ireland situation in London.

Example Output Good Friday pact was agreed in 1998 # Good Friday pact was a peace pact # Good Friday pact set up a joint Catholic-Protestant administration in Northern Ireland # Good Friday pact was mediated by the US # Irish Republican Army increased activity # Irish PM Ahern called to end violence # Sinn Fein Adams called to end violence # Hope in Northern Ireland that the IRA will disarm # British PM Blair and Ahern will chair a review of the Northern Ireland situation in London

Testing Reference Summary Netherlands midfielder Wesley Sneijder has joined French Ligue 1 side Nice on a free transfer.
Output From GPT-3.5-Turbo Netherlands midfielder Wesley Sneijder has joined Nice # Sneijder was a free transfer # Nice is a French Ligue 1 side
Output From GPT-4 Netherlands midfielder Wesley Sneijder # Sneijder joined French Ligue 1 side Nice # Joined on a free transfer

A.3 Extrinsic Evaluation Details

System-level correlation assesses the metric's ability to compare different summarization systems. This is denoted as K and measures the correlation between human scores (h), the metric (m), and the generated summaries (s) for N examples across S systems in the meta-evaluation dataset. The system-level correlation is then defined as:

$$K_{m,h}^{sys} = K([\frac{1}{N}\sum_{i=1}^{N} m(s_{i1}), ..., \frac{1}{N}\sum_{i=1}^{N} m(s_{iS})], \\ [\frac{1}{N}\sum_{i=1}^{N} h(s_{i1}), ..., \frac{1}{N}\sum_{i=1}^{N} h(s_{iS})])$$

Summary-level correlation assesses the metric's ability to compare summaries produced by different systems for a common document(s). The summary-level correlation is then defined as:

$$K_{m,h}^{sum} = \frac{1}{N}\sum_{i=1}^{N} K([m(s_{i1}), ..., m(s_{iS})], \\ [h(s_{i1}), ..., h(s_{iS})])$$

A.4 Dataset Details

The TAC08 dataset includes 96 examples and outputs from 58 systems, while TAC09 contains 88 examples and outputs from 55 systems. Both datasets contain multiple reference summaries for each example, as well as the corresponding SCU annotations.

The REALSumm dataset contains 100 test examples from the CNN/DM dataset (Hermann et al., 2015) and 25 system outputs. The SCUs are labeled by the authors and SCU-presence labels are collected using Amazon Mechanical Turk (AMT).

PyrXSum (Zhang and Bansal, 2021) includes 100 test examples from the XSum dataset (Narayan et al., 2018), which contains short and abstractive summaries. Similar to REALSumm, the SCUs are manually labeled by the authors and SCU-presence labels are collected for summaries generated by 10 systems through AMT.

A.5 Human annotated evaluation

The text units of each example were analyzed regarding Well-formedness, Descriptiveness and Hallucination. For each dimension, we classified it

[4] https://openai.com/blog/openai-api

into one of three categories based on the evaluator's satisfaction with the system's output. These categories ranged from "1 - Unhappy with system output", "2 - implying dissatisfaction or a less than satisfactory result", to "3 - Okay with system output (3)". Below we denote ASCU for approximated summary content unit (e.g., SGUs_4, SMUs, STUs or SCUs) and provide a detail definition for each evaluation category:

- Well-formedness (surface quality)
  - 1: Many ASCUs are are not concise English sentences
  - 2: Some ASCUs are not concise English sentences
  - 3: Almost all or all ASCUs are concise English sentences
- Descriptiveness (meaning quality I)
  - 1: Many meaning facts of the summary have not been captured well by the ASCUs
  - 2: Some meaning facts of the summary have not been captured by the ASCUs
  - 3: Almost every or every meaning fact of the summary has been captured by a ASCU
- Hallucination (meaning quality II)
  - 1: Many ASCUs describe meaning that is not grounded in the summary
  - 2: There is some amount of ASCUs that describes meaning that is not grounded in the summary
  - 3: Almost no or no ASCU describes meaning that is not grounded in the summary

In the following we show two examples of the reference summaries and the corresponding ASCUs from PyrXSum and RealSumm, respectively:

- **Reference summary:** West Ham say they are "disappointed" with a ruling that the terms of their rental of the Olympic Stadium from next season should be made public.
- **SCUs:** West Ham are "disappointed" with a ruling # The ruling is that their rental terms should be made public # West Ham will rent the Olympic Stadium from next season
- **SMUs:** West Ham say they are disappointed by the ruling that their terms of rental for the Olympic Stadium next season should be public # The ruling that the terms of West Ham's Olympic Stadium rental next season should be public was disappointing # West Ham rent the Olympic Stadium # West Ham will rent the Olympic Stadium next season
- **SGUs_4:** West Ham is disappointed with a ruling # Terms of their Olympic Stadium rental should be made public # Olympic Stadium rental starts next season
- **STUs:** West Ham say they are "disappointed" with a ruling that the terms of their rental of the Olympic Stadium from next season should be made public # They are "disappointed" with a ruling that the terms of their rental of the Olympic Stadium from next season should be made public # should made public