Revisiting text decomposition methods for NLI-based factuality scoring of summaries

John Glover[1] **Federico Fancellu**[1] **Vasudevan Jagannathan**[1]
Matthew R. Gormley[1,2] **Thomas Schaaf**[1]

[1]3M Health Information Systems
{jglover,ffancellu,juggy,tschaaf}@mmm.com

[2]Carnegie Mellon University
mgormley@cs.cmu.edu

Abstract

Scoring the factuality of a generated summary involves measuring the degree to which a target text contains factual information using the input document as support. Given the similarities in the problem formulation, previous work has shown that Natural Language Inference models can be effectively repurposed to perform this task. As these models are trained to score entailment at a sentence level, several recent studies have shown that decomposing either the input document or the summary into sentences helps with factuality scoring. But is fine-grained decomposition always a winning strategy? In this paper we systematically compare different granularities of decomposition – from document to sub-sentence level, and we show that the answer is no. Our results show that incorporating additional context can yield improvement, but that this does not necessarily apply to all datasets. We also show that small changes to previously proposed entailment-based scoring methods can result in better performance, highlighting the need for caution in model and methodology selection for downstream tasks.

1 Introduction

With improvements largely driven by recent advances in pre-trained language models (Vaswani et al., 2017; Radford et al., 2018; Lewis et al., 2020), modern abstractive summarization models are capable of producing summaries that are both fluent and coherent. However, they are still prone to various forms of "hallucination", generating statements that are not supported by the input text (Cao et al., 2018; Maynez et al., 2020). This has lead to a growing interest in being able to accurately measure the degree to which machine-generated output is non-factual (Falke et al., 2019; Kryscinski et al., 2020; Pagnoni et al., 2021; Laban et al., 2022).

In factuality scoring and other closely related tasks such as fact verification (Vlachos and Riedel, 2014; Thorne et al., 2018), the objective is to assess whether or to what degree the claims in a given text can be supported by other "evidence" texts. Given this setup, previous work has drawn a parallel with the task of Natural Language Inference (NLI), which has a similar goal of determining whether the meaning of one text can be inferred (entailed) from another (Dagan et al., 2006). As a consequence, models trained on large NLI datasets (Bowman et al., 2015; Williams et al., 2018; Nie et al., 2020) have often been successfully repurposed for the task of detecting factual inconsistencies in machine-generated summaries (Falke et al., 2019; Kryscinski et al., 2020; Maynez et al., 2020; Zhang and Bansal, 2021). It is now common that high-performance NLI models are trained on a combination of NLI and fact verification datasets (Nie et al., 2020; Schuster et al., 2021).

One way to repurpose NLI models for factuality scoring is to use the full text of the input and summary as the premise and hypothesis respectively, then take the factuality score to be a function of the model output distribution. However, NLI models are usually trained with sentence pairs as input, and can suffer performance degradation with the longer contexts that arise in summarization (Laban et al., 2022; Honovich et al., 2022). Worse yet, the majority of modern NLI models are based on architectures such as the Transformer (Vaswani et al., 2017) that use fixed-length input sizes, and it may not be possible for a full document and summary pair to fit into this context.

Another approach to NLI-based factuality scoring is grounded in the idea of first decomposing the input text into finer levels of granularity, followed by a later score aggregation step. Falke et al. (2019) proposed a scoring method based on sentence level decomposition, but concluded that the NLI models at the time were not robust enough for the task. However, recently both Schuster et al. (2022) and Laban et al. (2022) have shown that variations on this decomposition-based strategy, in combination

Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM), pages 97 - 105

with the improved performance of modern NLI models, can produce systems that perform well at the task of detecting factual inconsistencies in generated summaries.

In this work we revisit existing studies of NLI-based factuality scoring and perform a systematic comparison of input-summary decomposition methodologies at different levels of granularity – from document to sub-sentence level. We show that contrary to previous findings, adding more context to the premise (the source document) can sometimes outperform approaches based on a more fine-grained decomposition. We also find that small changes to the factuality scoring function can lead to a substantial increase in performance, but that model performance does not necessarily generalize across benchmarks that use different metrics (even when applied to the same underlying data). Our results highlight the need for caution and additional evaluation when selecting a model and methodology for downstream tasks.

2 Decomposition-based factuality scoring

In this work we are primarily concerned with *referenceless* factuality scoring of document summaries. To do so, we therefore require a function from an input $(document, summary)$ pair to a score value $Z \in \mathbb{R}$. NLI models typically learn a function that maps a pair of input text strings (X_{prem}, X_{hyp}), commonly referred to as the *premise* and *hypothesis*, to a probability distribution over the output classes *entailment*, *neutral*, or *contradiction*. One simple way to repurpose NLI models for factuality scoring is with $(document, summary)$ as (X_{prem}, X_{hyp}), and to take the score Z to be some function $f_Z(p_e, p_n, p_c)$ over the probability values given for entailment (p_e), neutral (p_n), or contradiction (p_c)[1]. We experiment with three decomposition-based scoring methods, described in the following sections.

2.1 SummaC

The SummaC models proposed by Laban et al. (2022) decompose the document and summary into sentences. A document is split into M sentences labelled $D_1, \ldots, D_M$, and a summary into N sentences $S_1, \ldots, S_N$. Each (D_m, S_n) combination is then passed through an NLI model, with scores computed using a function of the output probabilities. This decomposition results in an $M \times N$ score matrix for each $(document, summary)$. Laban et al. (2022) describe two model classes, which differ in how they process the score matrix to create a final factuality score for a summary:[2]

SUMMAC ZERO-SHOT (SC_{ZS}): each summary sentence is first scored by taking the maximum score value computed against any of the document sentences ($\max$ over each column in the $M \times N$ matrix). These summary sentence scores are then averaged to compute the final score.

SUMMAC CONVOLUTION (SC_{Conv}): the pair matrix is converted to a histogram by placing the score values into evenly spaced bins, then the resulting matrix is passed through a 1-D convolutional layer. We refer the reader to Laban et al. (2022) for further details.

We observe that although Laban et al. (2022) indicate that the scoring function f_Z that they use is given by $f_Z = p_e$, the default parameters in their publicly available code[3] describe $f_Z = p_e - p_c$. We compare these two variants of the score function $\mathbf{f}_Z$ in § 3.1.

2.2 SENTLI

Similarly to Laban et al. (2022), Schuster et al. (2022) propose a factuality scoring model that assigns a score for each summary sentence S_n according to the maximum score across all $(D_{1,\ldots,M}, S_n)$ pairs. Each (D_m, S_n) is scored using a custom NLI model based on T5 (Raffel et al., 2020) and fine-tuned on a combination of the SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018), ANLI (Nie et al., 2020), FEVER (Thorne et al., 2018) and VitaminC (Schuster et al., 2021) datasets.

Final scores are either the average score for all $S_{1,\ldots,N}$ in an aggregation method referred to as "soft aggregation", or the *minimum* score across $S_{1,\ldots,N}$ in their "hard aggregation" method. In addition, Schuster et al. (2022) propose an extension to this approach called "retrieve and rerank" ($\text{SENTLI}_{\text{RR}}$). Here they again first score all (D_m, S_n) using an NLI model. For each S_n, the top-K D_m are selected according to both the entailment and contradiction scores p_e and p_c. The NLI

[1] We note that generally the NLI models are not well-calibrated, and so these probability values may not necessarily have semantically meaningful interpretations, but empirically they can often be used directly in this manner.

[2] These models are agnostic to the particular NLI model being used for scoring, but the best performing model in the paper uses a version of ALBERT (Lan et al., 2020) fine-tuned on a combination of MNLI and VitaminC.

[3] https://github.com/tingofurro/summac

model is then presented with the same hypothesis S_n, together with a concatenation of the top-K entailing and contradicting sentences, with the output used to create the final score that S_n. For further details we refer the reader to Schuster et al. (2022).

2.3 Summarization Content Units (SCU)

Following Nenkova and Passonneau (2004) and Shapira et al. (2019), we take decomposition a step further and segment each summary into smaller units called *Summarization Content Units* (SCUs). In its original formulation, SCUs are hand-crafted short spans of text describing a single fact contained in one or more reference summaries[4]. As our evaluation data is not manually annotated with SCUs, we follow the method in Zhang and Bansal (2021), where the authors show that SCUs can be approximated using heuristics applied to the output of a Semantic Role Labeler. However, whereas these methods apply to *reference-based* evaluation of summaries, in the absence of human reference, here we adapt them to fit the reference*less* evaluation scenario. We refer to our method of decomposition and scoring with SCUs as SCU_{ZS}, and describe the details of the method in Appendix D.

3 Experiments and evaluation

We evaluate the performance of our models on the SummaC benchmark (Laban et al., 2022), which comprises of six datasets for summary inconsistency detection: CoGenSumm (CGS) (Falke et al., 2019), XSumFaith (XSF) (Maynez et al., 2020), Polytope (PT) (Huang et al., 2020), FactCC (FCC) (Kryscinski et al., 2020), SummEval (SE) (Fabbri et al., 2021), and FRANK (FR) (Pagnoni et al., 2021). Evaluation is standardized by casting each task as binary classification, and then measuring performance using balanced accuracy. As the NLI-based factuality scoring methods all output a scalar score value, we follow Laban et al. (2022) and tune thresholds separately for all methods and all datasets on the validation set, and report results using these threshold values on the test set. Although the FRANK dataset is part of SummaC, we also perform a separate evaluation of it using the original metrics of Pearson and Spearman correlations of the model output scores with (non-binary) human scores.

To assess the benefits of decomposing text for NLI-based factuality scoring, we compare the performance of the aforementioned decomposition methods with full text scoring, where either or both the source document or the summary has not been decomposed. We also test with a context length of several sentences, computed using a simplified version of the $\text{SENTLI}_{\text{RR}}$ method that we refer to as TOPK, as follows:

- First decompose the document and summary into individual sentences $(D_{1,\ldots,M}, S_{1,\ldots,N})$, and score all combinations using an NLI model.
- For each S_n, select the top-K sentences in $D_1, \ldots, D_M$ according to p_e.
- Concatenate these top-K sentences to form a new premise string.
- Run hypothesis S_n and the new premise through the NLI model, again taking p_e as the final score for S_n.
- Compute the final factuality score as the average over the scores for each S_n.

To split text into sentences we use spaCy (Honnibal et al., 2020). We note that Laban et al. (2022) used NLTK (Bird et al., 2009) for sentence-splitting, but this fails to correctly split sentences on some examples with bad punctuation (which are common in the FRANK dataset in particular[5]). In all experiments, unless otherwise specified we use the NLI model from Schuster et al. (2021) that is fine-tuned on a combination of Vitamin-C and MNLI datasets[6], which we refer to as VITC. For fair comparison with Laban et al. (2022), we set the maximum "full document" context for the premise to be 500 tokens.

3.1 Results

Our main results are summarized in Table 1, with SummaC results at the top and FRANK results at the bottom. In general, we find that factuality scoring using $f_Z = p_e$ has superior performance to $f_Z = p_e - p_c$, for all levels of input granularity, and for all evaluation metrics. We surpass both the original SC_{ZS}/SC_{Conv} and SENTLI/$\text{SENTLI}_{\text{RR}}$ SummaC results using SC_{ZS} with this scoring function. Further performance gains are also obtained from using additional context for the premise using TOPK, and we find that including the full document

[4] Example SCUs are given in Appendix D

[5] see Appendix A for details

[6] This is the best performing NLI model in Laban et al. (2022).

SummaC

System	f_Z	PG	HG	CGS	XSF	PT	FCC	SE	FR	Overall
SC_{ZS}				70.4∗	58.4∗	62.0∗	83.8∗	78.7∗	79.0∗	72.1∗
SC_{Conv}				64.7∗	66.4∗	**62.7∗**	89.5∗	81.7∗	81.6∗	74.4∗
SENTLI (soft)				79.3∗	59.3∗	52.4∗	89.5∗	77.2∗	82.1∗	73.3∗
$SENTLI_{RR}$ (soft)				79.6∗	62.7∗	52.8∗	86.1∗	78.5∗	80.4∗	73.3∗
$SENTLI_{RR}$ (hard)				**80.5∗**	64.2∗	55.1∗	83.3∗	79.7∗	78.4∗	73.5∗
SC_{ZS}	$p_e - p_c$	sent	sent	62.5	53.8	57.6	83.9	77.1	79.2	69.0
SC_{ZS}	p_e	sent	sent	76.8	65.6	57.6	**89.9**	79.7	81.3	75.1
SC_{ZS}	p_e	doc	doc	59.3	**69.9**	59.9	84.7	78.7	81.2	72.3
SC_{ZS}	p_e	TOPK	sent	79.7	67.3	56.9	89.4	81.8	81.4	76.1
SC_{ZS}	$p_e - p_c$	doc	sent	76.3	69.0	58.2	85.4	83.3	**82.6**	75.8
SC_{ZS}	p_e	doc	sent	76.2	69.8	61.7	84.6	**84.0**	82.0	**76.4**
SCU_{ZS}	p_e	TOPK	SCU	72.9	65.6	57.1	80.5	82.1	81.7	73.3
SCU_{ZS}	p_e	sent	SCU	71.4	63.4	55.0	77.0	80.0	81.4	71.4

FRANK

System	f_Z	PG	HG	Pearson ρ	p-val	Spearman r	p-val
FactCC				0.20∗	0.00∗	0.30∗	0.00∗
BertScore P Art				0.30∗	0.00∗	0.25∗	0.00∗
SC_{ZS}	$p_e - p_c$	sent	sent	0.32	0.00	0.26	0.00
SC_{ZS}	p_e	sent	sent	0.35	0.00	**0.36**	0.00
SC_{ZS}	p_e	doc	doc	0.31	0.00	0.25	0.00
SC_{ZS}	p_e	TOPK	sent	**0.37**	0.00	0.34	0.00
SC_{ZS}	$p_e - p_c$	doc	sent	0.30	0.00	0.26	0.00
SC_{ZS}	p_e	doc	sent	0.34	0.00	0.29	0.00
SCU_{ZS}	p_e	TOPK	SCU	0.36	0.00	0.30	0.00
SCU_{ZS}	p_e	sent	SCU	0.36	0.00	0.34	0.00

Table 1: Test set results for SummaC and FRANK. Results marked "∗" are taken from prior work, the rest are from our implementations. "PG" and "HG" are the premise and hypothesis levels of granularity respectively. Sentences in our implementations are split using spaCy.

context in the premise performs best of all, in contradiction to previous findings on this benchmark[7]. We see no additional performance benefit in going below the sentence level and using SCUs on these benchmarks, but the SCU decomposition does perform competitively across both benchmarks.

None of our variations achieve similar performance to the published SC_{ZS} results, either performing better or worse depending on whether f_Z is p_e or $p_e - p_c$ respectively. We believe that this discrepancy is due to the fact that the published SC_{ZS} results use classification thresholds that are tuned on the test set[8] rather than validation set.

On FRANK, we find that there is no single method that performs best across both correlation metrics, TOPK having the highest Pearson correlation, and the sentence level SC_{ZS} the highest Spearman correlation. It is notable however that the larger premise context granularity DOC-SENT is not as strong when using the original FRANK metrics as it is on SummaC, highlighting the need to be careful when comparing methods using different metrics, even on the same underlying data.

4 Conclusion

In this work we revisited prior findings that the best way to use NLI models for factuality scoring of machine-generated summaries is to first decompose the input to sentence level, score using NLI, then aggregate the sentence level scores to produce a document-level score. Contrary to prior work, we find that there is no single optimal level of decomposition that performs best across all tasks and evaluation metrics. We showed that in general, sentence level decomposition is preferable for the summary/hypothesis side of the NLI input, but on the premise side recent models such as VITC often benefit from having longer input contexts available when scoring. We also show that for the six datasets in the SummaC benchmark, there is still considerable variation in the performance of our methods both across the individual datasets, and also within different metrics on the same dataset.

[7] In Appendix B we show that some of these findings appear to be unique to this particular choice of NLI model.

[8] Confirmed via correspondence with Laban et al. (2022).

Limitations

Although we evaluate our methods across six different datasets, all are broadly from the same narrow domain, namely English news articles. We also note that despite the methods in Section 2 being agnostic to the choice of the NLI model that is used for scoring, there can be considerable degradation in the performance of methods that use longer premise contexts with some NLI models. More details can be found in Appendix B.

References

Stephen Bird, Edward Loper, and Ewan Klein. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the Original: Fact-Aware Neural Abstractive Summarization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. In *Machine learning challenges. Evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating Summarization Evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking Generated Summaries by Correctness: An Interesting but Challenging Application for Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Florence, Italy. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating Factual Consistency Evaluation. ArXiv:2204.04991 [cs].

Dandan Huang, Leyang Cui, Sen Yang, Guangsheng Bao, Kun Wang, Jun Xie, and Yue Zhang. 2020. What Have We Achieved on Text Summarization? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 446–469. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the Factual Consistency of Abstractive Text Summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346. Association for Computational Linguistics.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC : Re-Visiting NLI-based Models for Inconsistency Detection in Summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *8th International Conference on Learning Representations, ICLR 2020*, Addis Ababa, Ethiopia.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*. ArXiv: 1907.11692.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On Faithfulness and Factuality in Abstractive Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919. 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. Association for Computational Linguistics.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding Factuality in Abstractive Summarization with FRANK: A Benchmark for Factuality Metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving Language Understanding by Generative Pre-Training. *OpenAI Technical Report*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Tal Schuster, Sihao Chen, Senaka Buthpitiya, Alex Fabrikant, and Donald Metzler. 2022. Stretching Sentence-pair NLI Models to Reason over Long Documents and Clusters. *arXiv:2204.07447 [cs]*.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get Your Vitamin C! Robust Fact Verification with Contrastive Evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643. 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*, pages 682–687, Minneapolis, Minnesota. Association for Computational Linguistics.

Peng Shi and Jimmy Lin. 2019. Simple BERT Models for Relation Extraction and Semantic Role Labeling. *arXiv:1904.05255 [cs]*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for Fact Extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Andreas Vlachos and Sebastian Riedel. 2014. Fact Checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 18–22, Baltimore, MD, USA. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

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. Association for Computational Linguistics.

A Performance variations with different sentence-splitting methods

Table 2 describes how the performance of the SummaC Zero-Shot factuality scoring method varies based on whether NLTK or spaCy is used for sentence-splitting. All methods use the VITC NLI model. On SummaC, we see that using spaCy results in a slight improvement overall, whether our scoring function is $f_Z = pe$ or $f_Z = p_e - p_c$. We note that this is true for the FRANK dataset when scored using the SummaC balanced accuracy metric. However, on the FRANK dataset with the original metrics, we mostly see the opposite effect; using NLTK results in higher Pearson correlations for both scoring functions, and a higher Spearman for $f_Z = p_e - p_c$. Notably, the 0.39 Pearson correlation for SC_{ZS} at sentence level granularity using NLTK is the highest score that we obtain on this benchmark.

However, the results on Frank seem to be partly an artifact of inaccurate sentence-splitting by NLTK resulting in $(premise, hypotheis)$ pairs that are in fact at much larger levels of granularity than the intended sentence level, making this result

SummaC

System	f_Z	Splitter	CGS	XSF	PT	FCC	SE	FR	Overall
SC_{ZS}	$p_e - p_c$	NLTK	61.9	53.7	56.3	83.4	78.2	78.4	68.6
SC_{ZS}	$p_e - p_c$	spaCy	62.5	53.8	57.6	83.9	77.1	79.2	69.0
SC_{ZS}	p_e	NLTK	75.6	65.3	60.4	89.5	80.1	79.1	75.0
SC_{ZS}	p_e	spaCy	76.8	65.6	57.6	89.9	79.7	81.3	75.1

FRANK

System	f_Z	Splitter	Pearson ρ	p-val	Spearman r	p-val
SC_{ZS}	$p_e - p_c$	spaCy	0.27	0.00	0.23	0.00
SC_{ZS}	$p_e - p_c$	NLTK	0.32	0.00	0.26	0.00
SC_{ZS}	p_e	spaCy	0.35	0.00	0.36	0.00
SC_{ZS}	p_e	NLTK	0.39	0.00	0.34	0.00

Table 2: Performance differences on SummaC and FRANK test sets based on choice of sentence-splitting method. All methods use sentence level granularity for both premise and hypothesis. For SummaC all methods use thresholds selected using the validation set.

difficult to interpret. The following is an example of a passage of text taken verbatim from the FRANK validation set:

> Thousands attended the early morning service at Hyde Park Corner and up to 400 people took part in a parade before the wreath-laying at the Cenotaph.Anzac Day commemorates the first major battle involving Australian and New Zealand forces during World War One.A service was also held at Westminster Abbey.The national anthems of New Zealand and Australia were sung as the service ended.

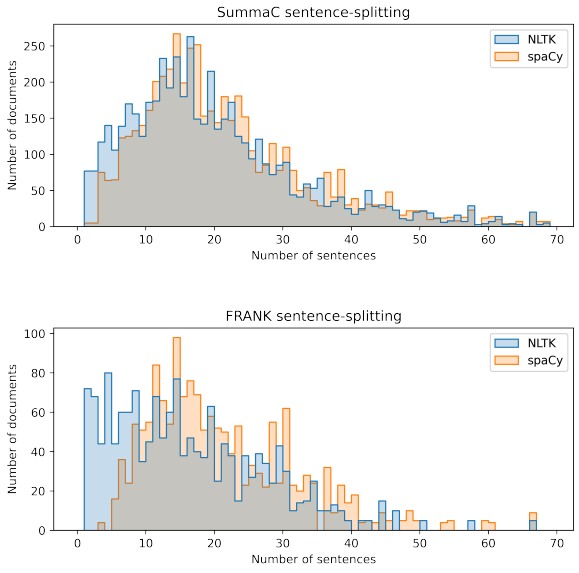

Figure 1: The number of sentences produced by NLTK and spaCy on SummaC and FRANK.

We note that in this example there is no space after the fullstops, which causes NLTK's parser to break. NLTK produces 1 sentence for this block of text, while spaCy produces 4 as we would expect. This issue is relatively frequent in the FRANK dataset. Figure 1 shows the distributions of the number of sentences produced by NLTK versus spaCy for all of the documents in both SummaC and FRANK, with statistics given in Table 4. We see that spaCy produces more sentences generally, with the difference being more pronounced on the FRANK dataset.

B Performance variations with different NLI models and levels of granularity

In Table 3 we investigate how changing the level of decomposition effects the performance of two additional NLI models. Notably with both of these models, scoring using the full document as the premise is significantly worse than either sentence level decomposition, TopK, or SCU, emphasizing that the results in Table 3 are highly dependent on the performance of the VitC NLI model. TopK and sentence level both perform reasonably well with these NLI models however, with the former being the best method to use on SummaC with $\text{Roberta}_{\text{ANLI}}$ and the latter the best with $\text{Roberta}_{\text{MNLI}}$. Again, we see no performance benefit when going to the SCU level.

C SCU examples

Two example one-line summaries, along with two extracted SCUs are shown below. Colors indicate which parts of the generated summaries the SCUs are extracted from.

> Summary_1: In 1998 two Libyans indicted in 1991 for the Lockerbie bombing were still in

SummaC

System	PG	HG	CGS	XSF	PT	FCC	SE	FR	Overall
ROBERTA$_{MNLI}$	doc	sent	58.1	56.2	52.9	62.5	57.0	66.2	58.8
ROBERTA$_{MNLI}$	TOPK	sent	61.5	63.3	60.0	81.5	75.1	76.4	69.6
ROBERTA$_{MNLI}$	sent	sent	75.2	61.3	59.2	90.7	80.1	79.5	74.3
ROBERTA$_{MNLI}$	TOPK	SCU	66.3	62.0	51.5	74.8	73.2	76.2	67.3
ROBERTA$_{MNLI}$	sent	SCU	71.6	65.1	53.9	81.9	77.0	80.0	71.6
ROBERTA$_{ANLI}$	doc	sent	53.5	62.9	55.8	62.3	59.6	69.5	60.6
ROBERTA$_{ANLI}$	TOPK	sent	77.3	65.4	58.4	82.4	78.4	76.9	73.2
ROBERTA$_{ANLI}$	sent	sent	73.1	61.2	59.6	87.6	74.1	80.2	72.7
ROBERTA$_{ANLI}$	TOPK	SCU	74.5	64.3	59.2	82.1	77.8	77.9	72.6
ROBERTA$_{ANLI}$	sent	SCU	70.8	64.4	55.8	79.8	76.7	81.0	71.4

FRANK

System	PG	HG	Pearson ρ	p-val	Spearman r	p-val
ROBERTA$_{MNLI}$	doc	sent	0.16	0.00	0.11	0.00
ROBERTA$_{MNLI}$	TOPK	sent	0.23	0.00	0.21	0.00
ROBERTA$_{MNLI}$	sent	sent	0.27	0.00	0.27	0.00
ROBERTA$_{MNLI}$	TOPK	SCU	0.23	0.00	0.21	0.00
ROBERTA$_{MNLI}$	sent	SCU	0.26	0.00	0.27	0.00
ROBERTA$_{ANLI}$	doc	sent	0.05	0.04	-0.04	0.16
ROBERTA$_{ANLI}$	TOPK	sent	0.27	0.00	0.30	0.00
ROBERTA$_{ANLI}$	sent	sent	0.27	0.00	0.32	0.00
ROBERTA$_{ANLI}$	TOPK	SCU	0.24	0.00	0.23	0.00
ROBERTA$_{ANLI}$	sent	SCU	0.27	0.00	0.29	0.00

Table 3: Performance differences on SummaC and FRANK test sets based on choice of NLI model and level of granularity. For SummaC all methods use thresholds selected using the validation set. Sentences are split using spaCy. ROBERTA$_{MNLI}$ is the NLI model from Liu et al. (2019), and ROBERTA$_{ANLI}$ is from Nie et al. (2020).

Libya.

Summary$_2$: Two Libyans were indicted in 1991 for blowing up a Pan Am jumbo jet over Lockerbie, Scotland in 1988.

SCUs: [two Libyans were officially accused of the Lockerbie bombing, the indictment of the two Lockerbie suspects was in 1991]

D SCU-based decomposition details

To create SCUs for a passage of text, we first split it into sentences using spaCy. We then pass each sentence through co-reference resolution (Joshi et al., 2020), and then finally we create SCUs using the method based on Semantic Role Labeling (SRL) (Shi and Lin, 2019) described in Zhang and Bansal (2021). We use the publicly available code from Zhang and Bansal (2021)[9] for both the co-reference resolution and SRL-based SCU generation.

[9] https://github.com/ZhangShiyue/Lite2-3Pyramid, the authors refer to their SRL-based SCUs as *Semantic Triplet Units* (STUs).

To score a $(document, summary)$ pair, we experimented with decomposing either the document, the summary, or both into SCUs. Here we describe the two variations that performed best on initial validation experiments. The first scores summary SCUs against document sentences, and the second scores summary SCUs using longer passages of text from the document as context.

D.1 SENT-SCU

This method is the most similar conceptually to SCZS.

- First decompose the document and summary into individual sentences $(D_{1,\ldots,M}, S_{1,\ldots,N})$, and then further decompose each S_n into SCUs $S_{SCU_1}, \ldots, S_{SCU_J}$.
- Score all (D_m, S_{SCU_j}) combinations using an NLI model, and $f_Z = p_e$.
- The score for each S_{SCU_j} is taken to be the maximum over the $(D_1, \ldots, D_M, S_{SCU_j})$ pairs.
- For each S_n, average over the scores for $S_{SCU_1}, \ldots, S_{SCU_J}$ to calculate a score for

SummaC		
	NLTK	spaCy
Mean	20.6	22.4
Std. dev.	16.4	18.0
25^{th} %	11.0	12.0
50^{th} %	17.0	18.0
75^{th} %	26.0	28.0

FRANK		
	NLTK	spaCy
Mean	16.0	20.9
Std. dev.	11.3	11.3
25^{th} %	7.0	13.0
50^{th} %	14.0	18.0
75^{th} %	24.0	28.0

Table 4: Mean, standard deviation, and percentiles of the number of sentences produced by NLTK and spaCy on SummaC and FRANK.

that summary sentence, before averaging over the scores for each S_n to create the document factuality score.

D.2 TOPK-SCU

This is similar to the TOPK scoring method from § 3.

- First decompose the document and summary into individual sentences $(D_{1,\ldots,M}, S_{1,\ldots,N})$, and then further decompose each S_n into SCUs $S_{SCU_1}, \ldots, S_{SCU_J}$.
- Score all (D_m, S_{SCU_j}) combinations using an NLI model, and $f_Z = p_e$.
- For each S_{SCU_j}, we select the top-K sentences in $D_1, \ldots, D_M$ according to $f_Z = p_e$, and concatenate them to form a new premise string.
- Hypothesis S_{SCU_j} is re-scored using the new premise string, using $f_Z = p_e$ as the score for S_{SCU_j}.
- For each S_n we then first average over the scores for $S_{SCU_1}, \ldots, S_{SCU_J}$ to calculate a score for that summary sentence, before averaging over the scores for each S_n to create the document factuality score.