Crowdsourcing Lightweight Pyramids for Manual Summary Evaluation

Ori Shapira[1], **David Gabay**[1], **Yang Gao**[2], **Hadar Ronen**[1],
Ramakanth Pasunuru[3], **Mohit Bansal**[3], **Yael Amsterdamer**[1], **and Ido Dagan**[1]

[1]Bar-Ilan University [2]UKP Technische Universität Darmstadt [3]UNC Chapel Hill
{obspp18, dawid.gabay}@gmail.com
gao@ukp.informatik.tu-darmstadt.de, hadarg@gmail.com
{ram, mbansal}@cs.unc.edu, {amstery, dagan}@cs.biu.ac.il

Abstract

Conducting a manual evaluation is considered an essential part of summary evaluation methodology. Traditionally, the Pyramid protocol, which exhaustively compares system summaries to references, has been perceived as very reliable, providing objective scores. Yet, due to the high cost of the Pyramid method and the required expertise, researchers resorted to cheaper and less thorough manual evaluation methods, such as Responsiveness and pairwise comparison, attainable via crowdsourcing. We revisit the Pyramid approach, proposing a lightweight sampling-based version that is crowdsourcable. We analyze the performance of our method in comparison to original expert-based Pyramid evaluations, showing higher correlation relative to the common Responsiveness method. We release our crowdsourced Summary-Content-Units, along with all crowdsourcing scripts, for future evaluations.

1 Introduction

Evaluating *content* quality of summaries is an integral part of summarization research. Measuring the performance of a summarization system can be done through either automatic or manual evaluation. An automatic evaluation, in practice working at the lexical level, provides an inexpensive means of measuring the validity of a system, both for system comparisons and for quick development cycle testing. Due to the shallowness of the automatic approaches, their reliability is often perceived as insufficient (Owczarzak et al., 2012; Chaganty et al., 2018). This calls for the more expensive manual evaluation, which employs human-in-the-loop protocols for assessment.

The Pyramid method (Nenkova and Passonneau, 2004) is a prominent manual evaluation methodology that is considered highly reliable for comparing summarization systems. It relies on a small set of manually-crafted reference summaries, out of which all summary content units (SCUs) are manually extracted. System summaries are then manually checked for coverage of each individual SCU, from which an overall system score is derived. The Pyramid evaluation method's reliability comes at a cost. It requires laborious manual work performed by annotators who must browse through non-trivial guidelines (Passonneau, 2006). Due to these drawbacks, it was only used in a few DUC and TAC (NIST, 2014, 2018) benchmarks.

Instead, summarization work in recent years has mostly employed simpler manual evaluation approaches, such as Responsiveness and pairwise comparison, which do not rely on reference summaries and can be attained via crowdsourcing. Yet, these methods are quite subjective, since evaluators need to provide only a single global judgment for the quality of a summary (or a pair of summaries). Such judgments are far more subjective than the Pyramid score, which is derived from many, more objective, local decisions, each judging independently the presence of an individual SCU. Indeed, it was shown that the above subjective crowdsourcing-based evaluation methods are not reliable enough to produce consistent scores across experiments (Gillick and Liu, 2010).

We propose a simplified crowdsourcable and reproducible version of the Pyramid method, that suggests appealing advantages over prior crowdsourcable evaluation methods. Like the original Pyramid, our method leverages the strong signal of the reference summaries and similarly bases its score on less subjective SCU judgments. In contrast to the original Pyramid, we rely on statistical sampling rather than exhaustive SCU extraction and testing, lowering overall cost. Empirically, our method correlates with the original Pyra-

Proceedings of NAACL-HLT 2019, pages 682–687
Minneapolis, Minnesota, June 2 - June 7, 2019. ©2019 Association for Computational Linguistics

mid scores better than the common Responsiveness method, and shows better stability.

2 Background: Manual Summary Evaluation

The Pyramid method (Nenkova and Passonneau, 2004) consists of two manual phases. The first phase is *pyramid creation*, performed once when a dataset is constructed, per each input topic to be summarized (either a single document or a set of documents). In this phase, experts exhaustively extract all *SCU contributors* ("mentions"), each being a text span describing an individual fact. SCU contributors are extracted from *several* reference summaries of the source text. Coreferring SCU contributors across reference summaries are then merged into a single *SCU*, which is given a representative label. Each SCU is then assigned a weight, equal to the number of reference summaries in which it was found, indicating its salience.

The second phase is *system evaluation*, performed over the summaries produced by the evaluated system. Each Pyramid SCU for the source text is manually checked for its presence in the given system summary, whose Pyramid score is then computed as a normalized sum of the weights of the SCUs it contains. The overall system score is defined as the average Pyramid score over all its evaluated summaries. Although certain normalization variants attempt to weigh in SCU precision, the score is essentially an absolute "recall-style" interpretation reflecting the system's ability to cover the content units found in the reference summaries. Such a fairly robust score allows, in principle, system comparison across experiments (Nenkova and Passonneau, 2004).

We note that due to the Pyramid method's reliability, some research has been carried out on simulating the Pyramid method as a fully automatic one (Yang et al., 2016; Hirao et al., 2018). The hope of such a line of work is to find an automatic evaluation method that is more reliable than the commonly used ones, by taking the reference summary *semantic* content into account. Despite these efforts, automated Pyramid evaluations did not make their way yet to mainstream summary evaluation practices, where variants of the ROUGE metric (Lin, 2004) still prevail. In any case, as this paper focuses on manual evaluation, we compare our results to those of the manual Pyramid.

The *Responsiveness* method, introduced in DUC 2003 (NIST, 2003), does not require reference summaries. Instead, human evaluators typically read both the source text and the system summary. They then assign a single subjective score on a Likert scale for the summary quality, often with respect to a topic statement or guiding question. Finally, compared systems are ranked by the average score of their summaries. This method naturally developed into a crowdsourcing task, and is now used frequently in some variants (Grusky et al., 2018; Paulus et al., 2018).

Another common crowdsourcable evaluation method is pairwise comparison (Gao et al., 2018; Falke et al., 2017; Fan et al., 2018): an evaluator is asked to judge which of two competing summaries of the same text is superior, usually while observing the source text. This protocol allows comparing only two systems at a time, where the superior is determined by the total votes over all input texts. The obvious disadvantage of the approach is the difficulty of comparing many systems, in the absence of absolute scores. Also, this method may tend to suffer from transitivity inconsistencies when comparing multiple system pairs (Gillick and Liu, 2010).

The lightweight crowdsourcable Pyramid version we propose aims to preserve the interpretability and relative objectiveness of the Pyramid scores. This could provide absolute scores for comparing multiple systems, which the pairwise method does not, in a more reliable manner than Responsiveness evaluation.

3 Our Lightweight Pyramid Method

Our Lightweight Pyramid method mimics the two phases of the original Pyramid protocol in a crowdsourced setting, with some adjustments.

Pyramid creation. The input for this phase is several reference summaries of a topic. Each reference is presented to two crowd workers, asking to extract eight SCU-like statements, yielding 16 potential SCUs per reference summary. The instructions guide workers to copy-and-paste extractions from the text, possibly modifying them to stand-alone sentences, that should (a) be brief and focused on a single fact; (b) capture important information; (c) rely solely on the text rather than general knowledge of the worker. Further, the statements should appear in different places in the text.

The copy-and-paste approach allows us to easily detect and filter duplicate statements extracted from the *same* reference by both annotators, which we identify via bag-of-lemmas cosine similarity. Further, too long sentences are filtered. In our experiments (see Section 4), we were left with an average of about 13 SCUs per reference summary. Then, we take the union of SCUs from all reference summaries, which yielded in our experiments 51 SCUs on average per topic, coming from four reference summaries. These SCUs are used to create tasks for the system evaluation phase.

Recall that in the original Pyramid, SCUs are exhaustively collected; then, coreferring SCUs *between* reference summaries are merged and weighted by the number of reference summaries from which they originate. In contrast, our method enables using a *sample* of SCUs for evaluation, out of the SCUs collected in this phase (we have sampled, for uniformity, 32 SCUs per topic). Further, it avoids posing the task of merging coreferring SCUs *across* references, which is difficult and error-prone, particularly when expected from crowd workers. Instead, we rely on the higher likelihood of a repeated fact to be included in our sample, possibly more than once. This implicitly increases the expected impact of repeated facts on our evaluation.

System evaluation. In this phase, a crowd worker is presented with a system summary and a fixed-sized small set of SCUs (we used sets of 16 SCUs). The worker is asked whether each SCU can be inferred from the system summary text. The guidelines advise workers to refrain from using general knowledge and to ignore minor content differences between the SCU and the system summary. Each SCU should be assessed by a few crowd workers, to ensure the stability of the results (in our experiments, each SCU was assigned for evaluation to 5 workers).

Scoring. Following common practice in crowdsourcing, we use techniques of filtering out noisy workers who had high disagreement with others (pairwise worker agreement < 0.5). Then, using the remaining answers, we take the majority vote for each SCU to decide whether it appears in the system summary.[1] We resolve ties with a "not present" default, as the more likely answer. We then compute the system summary score as the percentage of SCUs it matched out of the set of judged SCUs. A system's final score is its average score over all topics.

4 Experiments

Experimental setup. We used the DUC 2005 and 2006 multi-document summarization datasets (NIST, 2014), which contain expert evaluations for both Pyramid and Responsiveness. Each of the two datasets includes 20 document clusters, each pertaining to a target topic, with four reference summaries and 25 (2005) or 22 (2006) system summaries per topic. All summaries are 250 words long. On average, 105 weighted SCUs were extracted, by experts, for each topic. In comparison, our setup gathers 32 sampled crowdsourced unweighted SCUs.

As suggested in Dang (2006) and Passonneau et al. (2006), the 2005 data tends to be easier to evaluate than the 2006 data, seemingly due to "less natural" document clusters with respect to practical summarization settings. Passonneau et al. (2006) show that the document sets in 2005 were overall more difficult for systems to summarize, as reflected by a lower average Pyramid score across all systems. The 2005 topics are more complex as they yield fewer general, context-independent SCUs. For example, as Dang (2006) indicates, there are more topics that had a relatively large number of specific named entities. Consequently, due to the topic hardness, Passonneau et al. (2006) indicate very few significant differences between overall system Pyramid scores, as evident by Tukey's HSD test. While 2006 systems can be divided into eight significantly different Pyramid score groups, in 2005 only two such groups emanate. Additionally, the guidelines and scoring method were slightly improved in 2006, relative to 2005. For these reasons, we focused on the 2006 dataset, fully annotating it, while utilizing half the topics, randomly chosen, from the 2005 data.

Using Amazon Mechanical Turk,[2] we qualified workers with over 5000 approved assignments and a 99% approval rate. We paid workers $0.50 per reference summary annotation assignment (generating 8 SCUs), yielding a total Pyramid creation cost of $48 (including fees) for the 2005 dataset (10 topics) and $96 for 2006 (20 topics). Pyramid

[1] In our experiments, we have also examined the option of using the average answer, which was significantly worse.

[2] https://www.mturk.com/

	Pearson (ρ_p)		Spearman (ρ_s)	
	Ours	Expert Resp.	Ours	Expert Resp.
2005	0.81	0.81	**0.79**	0.77
2006	**0.74**	0.60	**0.69**	0.40

Table 1: Correlations to the original Pyramid scores, for our crowdsourced method and for *expert* Responsiveness method, for DUC '05 and '06.

creation cost per topic is thus $4.8. For the system summary evaluation phase we split the 32 SCUs to two tasks of 16 SCUs each, in order to ensure that the crowdsourcing platform assigns each SCU to 5 distinct workers. We paid workers $0.45, and evaluated all 25 (2005) and 22 (2006) systems. The total benchmark evaluation cost was $1350 (including fees) for 2005 and $2376 for 2006, equaling $5.4 per system per topic, or $108 per system evaluation over all 20 topics.

We release[3] our SCU dataset for DUC 2005 and DUC 2006 as a complementary resource, accompanied by the HTML pages for our tasks on Amazon Mechanical Turk and processing and evaluation scripts. In the SCU dataset, we mark the SCUs we used in our experiments, including their grouping as tasks in the system evaluation phase. These enable future crowdsourced Pyramid evaluations of new systems on these datasets, as well as developing new datasets with crowdsourced pyramids.

Correlations with original Pyramid. We first assess our evaluation methodology by computing the correlation of its system scores (and rankings) to those of the original Pyramid. These are compared with the analogous correlations for the expert Responsiveness scores, available in the datasets. As seen in Table 1, our method produces better correlations, and substantially so on the more characteristic 2006 dataset. Importantly, notice that Responsiveness scores here were obtained by *experts*, and therefore the gap for crowdsourced Responsiveness is expected to be greater, further indicating the advantage of our method as a crowdsourcable approach.

Stability. As an additional assessment, we test the robustness of our method, in terms of its reproducibility. To that end, we reran the system evaluation phase on eight randomly chosen systems of the 2006 data, which enabled us to compare our results with those obtained by Gillick and Liu (2010) for crowdsourced Responsiveness for a similar setting (8 random systems of the 2006 dataset). Notably, the lightweight Pyramid obtained an average 10% relative change in overall system scores, whereas crowdsourced Responsiveness exhibited lower stability with an average of 24% relative change.

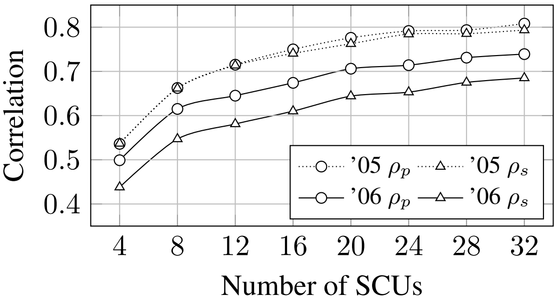

Figure 1: Average Pearson and Spearman correlations with Pyramid scores as a function of number of SCUs evaluated per topic, on the DUC '05 and '06 data.

Cost analysis. We analyze the impact of randomly reducing the various resources involved in our methodology, aiming to see whether overall cost might be reduced without harming correlation with the original Pyramid. The results below, reported as averages over 70 re-sampled iterations for each setting, suggest that such cost reductions would be harmful.

Number of workers. Reducing the number of workers per SCU judgment from five to three drops the correlations by about 8 points in 2006 and 6 points in 2005.

Number of SCUs. Figure 1 shows that correlation increases as a function of the number of judged SCUs per topic. The correlation improvement seems to stabilize around 32 SCUs.

Number of topics. Figure 2 presents the effect of the number of topics on which systems are evaluated, showing a steady correlation increase, which does not necessarily saturate at the number of 20 topics available in these datasets.

Qualitative analysis. To identify certain limitations of our methodology, we manually analyzed some "suspected" topics, for which either worker Krippendorff agreement or correlation with the original Pyramid was low. We noticed two interesting phenomena.

First, some topics seem inherently more difficult to evaluate, particularly for crowd workers. Such difficulty may be attributed to SCUs that are more difficult to assess or to less coher-

[3] https://github.com/OriShapira/LitePyramids

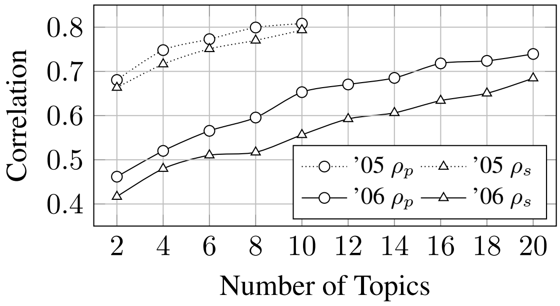

Figure 2: Average Pearson and Spearman correlations with Pyramid scores as a function of number of topics used for evaluation, on the DUC '05 and '06 data.

ent system summaries, due to the respective document set's complexity. Indeed, Passonneau et al. (2006) indicated that topic characteristics and annotator training experience effect evaluation quality. It seems worthwhile investigating, in future research, whether correlations improve by increasing further the overall number of topics, reducing the impact of the problematic ones.

Another possibility may be to filter out topics with low annotator agreement when computing systems' scores by the lightweight Pyramid method. We hypothesize that doing so might improve the reliability of this method, and hence increase its correlation with the original, expert-based, Pyramid method (when the latter is computed over all test topics). Indeed, in a preliminary test, we filtered out those 20% of the topics with lowest Krippendorff annotator agreement. This yielded a 6-point Spearman score increase (relative to the correlations reported in Table 1) when correlated with the original Pyramid ranking, as computed over the full set of topics. We note that while Figure 2 shows a slight decrease in average correlation when removing 4 random topics, removing specifically the 4 low-agreement topics seems to improves it notably. Further analysis might conclude that filtering problematic topics generically improves the reliability of the lightweight Pyramid method.

The second phenomenon observed among the difficult topics was that in some, the 32 sampled SCUs seem to miss important information, causing an unjustified degradation in system scores. In analogy to the variance in the number of SCUs in exhaustive Pyramids, it would be interesting to investigate methods for varying the sample size in our lightweight approach, based on some automatically detected parameters of topic complexity.

5 Conclusion and Future Work

To the best of our knowledge, our method is the first to mimic the reliable Pyramid method as an affordable crowdsourced procedure. Our experiments suggest that this lightweight Pyramid is more reliable than the common Responsiveness method. It also allows comparing multiple systems with absolute scores, which pairwise comparison does not.

Future work may improve correlation with the original Pyramid, or reduce annotation cost, by following our qualitative analysis and by reducing crowdsourcing noise (via qualification tests, enhanced guidelines, and post-processing result normalization (Hovy et al., 2013; Plank et al., 2014; Hosseini et al., 2012)). It would be appealing to investigate applying our methods to additional evaluation datasets, for which original Pyramid evaluations are not available for comparison. For example, addressing the CNN/DailyMail dataset (Nallapati et al., 2016) would involve testing single document summarization, utilizing a single reference summary per source text and addressing varying lengths of reference and system summaries.

The Pyramid method is mainly a measurement of recall, which thus also applies to our lightweight Pyramid; but other measurements for summary quality, such as precision, non-redundancy and grammaticality, may also be considered. In particular, it may be possible to extend our design of crowdsourcing tasks to supply indications for these complementary measurements as well.

Acknowledgments

We would like to thank the anonymous reviewers for their constructive comments, as well as Ani Nenkova for her helpful remarks. This work was supported in part by the Bloomberg Data Science Research Grant Program; by the German Research Foundation through the German-Israeli Project Cooperation (DIP, grants DA 1600/1-1 and GU 798/17-1); by the BIU Center for Research in Applied Cryptography and Cyber Security in conjunction with the Israel National Cyber Bureau in the Prime Minister's Office; by the Israel Science Foundation (grants 1157/16 and 1951/17); by DARPA Young Faculty Award YFA17-D17AP00022; and by the ArguAna Project GU 798/20-1 (DFG).

References

Arun Chaganty, Stephen Mussmann, and Percy Liang. 2018. The price of debiasing automatic metrics in natural language evalaution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 643–653.

Hoa Trang Dang. 2006. Overview of duc 2006. In *Proceedings of the document understanding conference*.

Tobias Falke, Christian M. Meyer, and Iryna Gurevych. 2017. Concept-map-based multi-document summarization using concept coreference resolution and global importance optimization. In *Proceedings of the 8th International Joint Conference on Natural Language Processing*, pages 801–811.

Angela Fan, David Grangier, and Michael Auli. 2018. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54. Association for Computational Linguistics.

Yang Gao, Christian M. Meyer, and Iryna Gurevych. 2018. April: Interactively learning to summarise by combining active preference learning and reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4120–4130. Association for Computational Linguistics.

Dan Gillick and Yang Liu. 2010. Non-expert evaluation of summarization systems is risky. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 148–151. Association for Computational Linguistics.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Tsutomu Hirao, Hidetaka Kamigaito, and Masaaki Nagata. 2018. Automatic pyramid evaluation exploiting edu-based extractive reference summaries. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4177–4186.

Mehdi Hosseini, Ingemar J Cox, Nataša Milić-Frayling, Gabriella Kazai, and Vishwa Vinay. 2012. On aggregating labels from multiple crowd workers to infer relevance of documents. In *European Conference on Information Retrieval*, pages 182–194. Springer.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with mace. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1120–1130.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

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

NIST. 2003. Duc 2003: Documents, tasks, and measures. `https://duc.nist.gov/duc2003/tasks.html`.

NIST. 2014. Document understanding conferences. `https://duc.nist.gov/`.

NIST. 2018. Text analysis conference. `https://tac.nist.gov/`.

Karolina Owczarzak, John M Conroy, Hoa Trang Dang, and Ani Nenkova. 2012. An assessment of the accuracy of automatic evaluation in summarization. In *Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, pages 1–9. Association for Computational Linguistics.

Rebecca Passonneau. 2006. Pyramid annotation guide: Duc 2006. `http://www1.cs.columbia.edu/~becky/DUC2006/2006-pyramid-guidelines.html`.

Rebecca Passonneau, Kathleen McKeown, Sergey Sigelman, and Adam Goodkind. 2006. Applying the pyramid method in the 2006 document understanding conference.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. *Sixth International Conference on Learning Representations*.

Barbara Plank, Dirk Hovy, and Anders Søgaard. 2014. Linguistically debatable or just plain wrong? In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (volume 2: Short Papers)*, volume 2, pages 507–511.

Qian Yang, Rebecca J. Passonneau, and Gerard de Melo. 2016. Peak: Pyramid evaluation via automated knowledge extraction. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI 2016)*. AAAI Press.