Towards an Automated Pointwise Evaluation Metric for Generated Long-Form Legal Summaries

Shao Min Tan
Thomson Reuters Labs
Landis + Gyr-Strasse 3
6300 Zug, Switzerland
shaomin.tan@tr.com

Quentin Grail
Thomson Reuters Labs
Landis + Gyr-Strasse 3
6300 Zug, Switzerland
quentin.grail@tr.com

Lee Quartey
Thomson Reuters Labs
3 Times Square
New York, NY 10036, USA
lee.quartey@tr.com

Abstract

Long-form abstractive summarization is a task that has particular importance in the legal domain. Automated evaluation metrics are important for the development of text generation models, but existing research on the evaluation of generated summaries has focused mainly on short summaries. We introduce an automated evaluation methodology for generated long-form legal summaries, which involves breaking each summary into individual points, comparing the points in a human-written and machine-generated summary, and calculating a recall and precision score for the latter. The method is designed to be particularly suited for the complexities of legal text, and is also fully interpretable. We also create and release a small meta-dataset for the benchmarking of evaluation methods, focusing on long-form legal summarization. Our evaluation metric corresponds better with human evaluation compared to existing metrics which were not developed for legal data.

1 Introduction

Generative text models, including large language models (LLMs), have made huge strides in performance in the last few years, and are now increasingly deployed in many domains in business and science. However, research on effective automated evaluation metrics for generated text has yet to catch up, and basic methodologies such as ROUGE (Lin, 2004) and others (see Section 2) are still used to judge the performance of new models. In the sub-field of text summarization, existing research on and meta-datasets for the evaluation of generated text summaries have focused mainly on shorter summaries consisting of a few sentences, while very little work has been done on long-form summaries (see the survey Koh et al., 2022).

Long-form abstractive summarization is a task that has particular importance in the legal domain. Legal documents such as court judgments (which are documents written by judges, detailing the background of a court case and the reasons for a ruling) are often many tens of pages long, and summaries of these can be several pages long. The UK Supreme Court, for instance, releases press summaries of 2-3 pages for the cases it decides (The Supreme Court of the United Kingdom, 2024).

Modern LLMs, with their long context windows, are a natural tool for automatically generating such summaries from the original legal document. There is a pressing need, therefore, for effective automated evaluation metrics for the resulting long-form summaries.

In this paper, we propose an automated method for the evaluation of long-form generated legal summaries, which involves breaking each summary into individual points, comparing the points in a human-written reference and machine-generated candidate summary, and calculating a recall or precision score. We call our method the *pointwise evaluation methodology*.

The idea of splitting summaries into discrete units to obtain reliable manual evaluation scores is well-known (Nenkova and Passonneau, 2004), and automated methods based on this idea have been explored (Liu et al., 2023b). Our proposed method expands upon previous work by: 1) adapting the methodology to be usable for long-form summaries and 2) using more advanced models to deal with the greater nuance and complexity of legal text.

To evaluate this method against existing ones, we also create and release a small meta-dataset for benchmarking evaluation methodologies for long-form legal summarization. To our knowledge, this is the first such dataset to be made available.

Proceedings of the Natural Legal Language Processing Workshop 2024, pages 129–142

2 Survey of Existing Approaches and Prior Work

2.1 Manual Evaluation Methodologies

Manual evaluation is considered to be the gold-standard for scoring the outputs of machine learning models. The following papers present systematic methods for collecting manual scores for generated text.

Pyramid (Nenkova and Passonneau, 2004) introduced a reliable method of obtaining human evaluations of generated summaries against a set of human-written reference summaries. The authors introduce the concept of Summarization Content Units (SCUs) — parts of a text that are no bigger than a single clause. The SCUs are manually extracted from each reference summary. If SCUs from multiple references have near-identical meanings, these are considered to be a single SCU, and this SCU is given a higher weight based on how many reference summaries it appears in.

The extracted SCUs are then used to objectively evaluate the candidate summaries. For each candidate summary, the human annotator determines which SCUs are contained within the candidate. The candidate is then assigned a score based on the weights of the SCUs it contains.

LitePyramid (Shapira et al., 2019) simplifies the Pyramid method by using statistical sampling rather than exhaustive SCU extraction and analysis, making the process less error-prone and more suited for crowdsourced workers. Instead of merging similar SCUs that appear in multiple documents, each SCU is considered individually during the annotation of candidate texts. A fact that is important will be repeated in different reference summaries and thus be "weighted" more strongly during the scoring process.

REALSumm (Bhandari et al., 2020) presents a meta-dataset for evaluation based on the CNN/Daily Mail dataset, produced by adapting the LitePyramid method to be used with only one human reference summary.

2.2 Automated Evaluation Metrics

Because human evaluation is time-consuming, automated evaluation metrics are often used to measure the quality of generated texts. While such automated methods are convenient, they may not correlate well with manual evaluation.

ROUGE-N (Lin, 2004) measures the overlap of n-grams between the reference and candidate texts.

ROUGE-L (Lin, 2004) measures the length of the longest common subsequence between the two texts, normalised by the length of one of the texts.

SEMScore (Aynetdinov and Akbik, 2024) measures the cosine similarity between the embeddings of the two texts.

BERTScore (Zhang et al., 2020) calculates the document similarity as a combination of the similarity between contextual BERT embeddings of individual tokens in the reference and candidate texts.

BARTScore (Yuan et al., 2021) is based on calculating the probability that the BART model would produce the candidate text given the reference text (or vice-versa).

FActScore (Min et al., 2023) calculates a factuality score for a generated text by breaking the generated text into atomic facts and calculating the percentage of facts supported by a reliable knowledge source. The authors focused on generated biographies.

AlignScore (Zha et al., 2023) measures the factual consistency between two texts using a general function of information alignment, developed using a variety of data sources from common NLP tasks.

A^2CU (Liu et al., 2023b) automates the LitePyramid method by 1) fine-tuning a T-Zero 3B model (Sanh et al., 2022) to extract content units from reference summaries, and 2) using a BERT-based (Devlin et al., 2019) Natural Language Inference (NLI) model to check whether each content unit is present in a generated candidate summary. The authors also developed a single-step metric (A^3CU). The authors trained and tested their models on short summaries (several sentences long) from the RoSE dataset (Liu et al., 2023a).

2.3 Meta-Datasets for Evaluation of Evaluation Metrics

TAC 2008 and TAC 2009 (Dang and Owczarzak, 2008): These datasets contain 100-word summaries of multiple documents, and include human evaluation of machine-generated summaries.

REALSumm CNNDM dataset (Bhandari et al., 2020): The authors created a meta-evaluation dataset based on the CNN/Daily Mail news summarization dataset. The gold summaries are an average of 3 - 4 sentences long.

RoSE dataset (Liu et al., 2023a): Meta-evaluation dataset of short-form summaries based on 3 datasets: CNN/Daily Mail, XSum (single-

sentence summaries of news articles), and SAMSum (dialogue summaries).

2.4 Text Summarization in the Legal Domain

Hachey and Grover, 2006 developed an extractive summarization method for UK court judgments using the rhetorical status of sentences.

In Shukla et al., 2022, the authors explored and evaluated various extractive and abstractive methods of summarizing legal case documents. They also performed a meta-evaluation study, and found that the results of several automated evaluation metrics (ROUGE and BERTScore) correlate poorly with human ratings. The authors did not release their meta-evaluation dataset.

3 Pointwise Evaluation Method

3.1 Introduction

We expand upon previous work by developing an interpretable, two-step evaluation methodology suited for legal text. The steps consist of:

1. Breaking the reference and candidate texts into individual points;

2. Determining, for each point in the reference text, whether there is a point in the candidate text that is saying the same thing (though it may be phrased differently), and vice-versa.

These steps can either be done manually (see Section 4) or using automated methods (see Section 5). A recall and precision score can then be calculated.

3.2 Differences from Existing Approaches

Our method differs from existing approaches in the following ways.

Granularity of Semantic Units

Nenkova and Passonneau, 2004 and Liu et al., 2023b break the text to be evaluated down into basic units of a single clause, as shown in the example in Figure 1 (a1). Basic units of this size can work well with news article summaries, which tend to concentrate on facts.

Legal documents such as court judgments, however, are more complex and often involve logical reasoning. The example in Figure 1 (a2) shows a legal sentence and what it would look like if broken into single-clause units. These units, however, are not a good representation of the original sentence in the context of a legal case. The first point, that the Court of Appeals disagreed with the High Court, is true but not useful without the additional information about which point they disagreed on. The Court of Appeals may well have agreed with the High Court on another legal issue while disagreeing on this one. The second point, that "the listings were not targeted at UK consumers", is stated as a fact, when in the original sentence it was the High Court's opinion. It is important to distinguish who says something in a legal case, because the parties and courts involved often have differing opinions.

We therefore use longer points as our base unit of text.

Handling Long-Form Summaries

The entailment models used in Liu et al., 2023b have been trained on short summaries and perform less well on long-form summaries.

In addition, long documents sometimes require greater contextual understanding of the document in order to determine whether two sentences are making the same point. Consider the example in Figure 1 (b). These two sentences are making the same point in the context of the court's reasoning, but one needs to know the context of the factors mentioned in the second sentence in order to be sure of this.

Handling Greater Nuance and Complexity of Legal Text

Because legal texts involve complex reasoning, it is a more difficult task to determine whether two sentences are making the same point in the context of a legal case. For example, consider the two sentences in Figure 1 (c). The two sentences are not making the same point, nor does either entail the other. However, the logic involved in the sentences is somewhat convoluted.

We therefore make use of more advanced models, such as state-of-the-art LLMs, which are better able to handle such nuanced reasoning tasks, especially when given examples in the prompt.

4 Meta-Dataset for Evaluation of Long-Form Legal Summaries

We create a small meta-dataset for the evaluation of evaluation methods for long-form legal summarization, consisting of 7 cases from the UK Supreme Court (UKSC) [1]. For each decided case, the UKSC writes and releases a 2-3-page-long press summary.

[1] Contains public sector information licensed under the Open Government License v3.0.

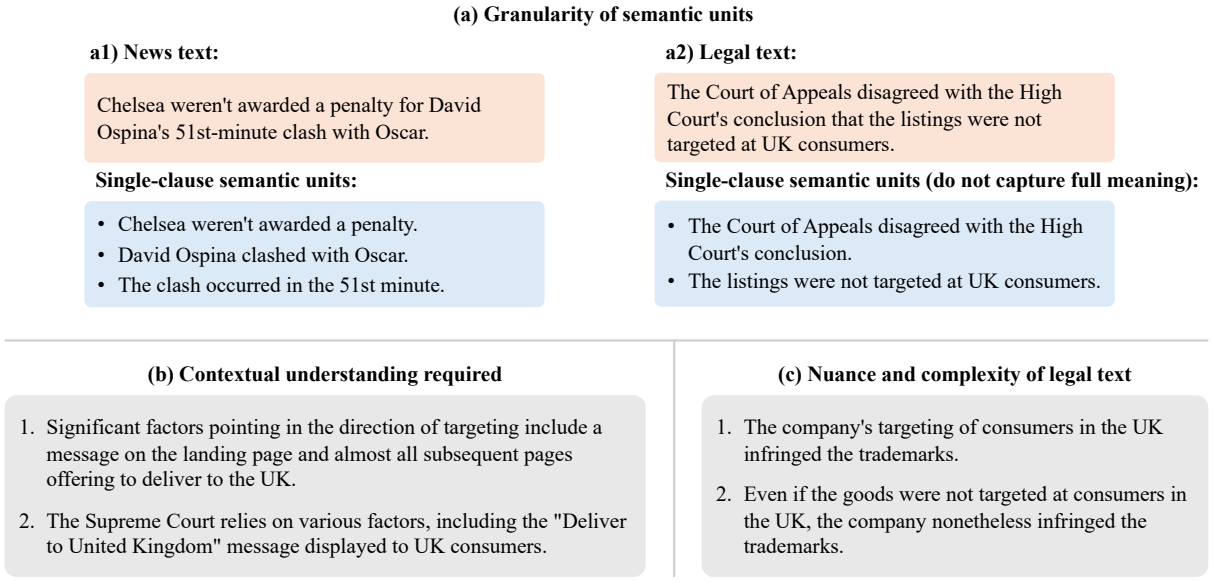

Figure 1: Examples of the nuances involved in legal language and how it differs from other types of text such as news. The examples are explained in Section 3.2. Example (a1) is adapted from Liu et al., 2023b.

We use this press summary as the human-written reference summary of the case.

For each court decision, we generate 5 LLM-written summaries, using different models (Claude 3 Opus and Sonnet, GPT-4o, and Titan Text G1 Premier) and prompts. We then use a variation of the LitePyramid method to create the meta-dataset, following the two-step procedure described in Section 3.1.

Step 1: Point Extraction. We manually break each summary (human-written reference and LLM-generated candidates) down into discrete units, which we call *points*. As explained in Section 3.2, these points tend to be more complex than the units used in previous work using the Pyramid method.

Step 2: Point Matching. For a given candidate summary, we step through each point in the reference summary, and find the best-matching candidate point (if any).

Further details can be found in Appendices A.1 and A.2.

4.1 Recall Score

To calculate the overall recall score of the candidate summary, we consider the percentage of reference points which have a matching candidate point, with a weighting scheme applied. The weighting scheme is described in Appendix A.3.

The pointwise evaluation method can also be used to obtain precision scores. However, in accordance with the literature, we concentrate on recall-based scoring when creating the dataset.

4.2 Dataset Split

Some of the automated methods discussed in the next section require training examples. We use 3 of the UKSC cases (each with 1 human-written and 5 LLM-generated summaries) as the training and validation set, and the remaining 4 cases (each with 1 human-written and 5 LLM-generated summaries) as the test set for calculating the performance of the automated methods.

5 Automating the Pointwise Evaluation Method

Each step in the two-step approach described in Section 3.1 can be automated.

5.1 Step 1: Point Extraction

In Step 1, we break the text into individual points. We investigate the following automated methods.

5.1.1 Fine-tuned T5 Model

Chen et al., 2023 introduced a semantic unit which the authors call *atomic expressions of meaning* or *propositions*. The authors fine-tuned a Flan T5-large model (Chung et al., 2024) on the FACTOID-WIKI dataset to extract propositions from an input passage. We have found that their model splits the text into longer segments than the content units presented in Nenkova and Passonneau, 2004 and Liu et al., 2023b, making it more suitable for legal text. To further increase the suitability of the model, we fine-tune the model on our dataset.

5.1.2 LLM Prompted with Examples

We prompt an LLM (Claude 3.5 Sonnet, which in our experience is the best-performing Anthropic model for similar tasks) to split a paragraph into individual points, giving it similar instructions to those in Appendix A.1. We also provide it with about 25 examples from the training dataset.

For both of these methods, we pass each paragraph of the summary separately through the model.

5.2 Step 2: Point Matching

In Step 2, we compare the points in the reference and candidate texts, to determine whether the same points exist in both. In other words, for each point in the reference text, we need to evaluate whether there is a point in the candidate text that is stating the same idea. This boils down to determining whether two sentences are making the same point in the context of the legal case (though they may be phrased differently).

We investigate the following methods (ranging from simple to complex) to automate this step.

5.2.1 Cosine Similarity

We calculate the embeddings of the two points, using the Sentence Transformers (Reimers and Gurevych, 2019) *all-mpnet-base-v2* model (the current top-performing Sentence Transformers model (Sentence Transformers, 2024)). Cosine similarity is then computed between these embeddings. All pairs passing a threshold are considered a match. The threshold is selected to optimize one of the downstream metrics (*reference point based F1*, described in Section 6.2) on the training set.

5.2.2 NLI Model

We use the NLI model from Liu et al., 2023b to check whether each reference point is present in the candidate summary. Their NLI model is a DeBERTa model (He et al., 2021) that has been fine-tuned on the RoSE dataset. Having been trained on short summaries, the NLI model does not perform well when presented with more than a few sentences, even though the model can theoretically take in longer text.

To adapt the model for long-form summaries, we make use of the paragraph structure of these summaries. For each reference point, we ask the NLI model whether each paragraph in the candidate summary entails the reference point. If at least one candidate paragraph entails it, the reference point is considered present in the candidate summary.

Additionally, we fine-tune this NLI model on our legal dataset.

5.2.3 LLM with Contextual Prompt and Examples

As described in Section 3.2, legal text involves greater nuance than news text, and sometimes requires understanding of the context of the whole legal case. To best handle these complexities, we use a state-of-the-art LLM to determine whether two points are making the same point in the context of a legal case. We chose Claude 3.5 Sonnet as our LLM, since in our experience it is the best-performing Anthropic model for similar tasks.

The LLM is provided with the following information in the prompt:

- Examples from other court cases
- The full reference summary, which gives the LLM the context of the court case
- The context for each of the two points to be compared (the context consists of the point itself and the preceding and following point, in order of appearance)
- The two points to be compared
- An explanation of what "making the same point" means in the context of a court case (shown in Appendix A.4)
- An instruction to the LLM to give a one-sentence explanation, followed by a binary rating ("Yes" or "No")

We experimented with two regimes for providing examples in the prompt:

Few-Shot Regime: We provide around 10 examples which include edge cases that are particularly tricky to distinguish. For each example, an explanation is provided, followed by the correct answer.

Many-Shot Regime: We carried out many-shot prompting (a concept explored in Agarwal et al., 2024) by providing the LLM with several hundred examples. To create these examples, we used 2 UKSC cases from our training set. For each case, we collated the few-shot-prompted LLM responses for the reference-candidate point pairs from one of the generated summaries. We extracted all the point pairs for which the LLM gave a true positive, false positive, or false negative response. Because there were many more true negatives, we extracted only a subset of these point pairs; for each reference point, we chose the candidate point with

the highest cosine similarity that was an LLM true negative. The incorrect LLM explanations (false positives or negatives) were then corrected by hand; in some cases the true positive explanations were also edited.

These examples were included in the many-shot prompt as follows: We first include the example court case summary, followed by a list of the example point pairs in that case. For each example point pair, we provided the hand-corrected explanation followed by the final "Yes" or "No" answer.

Pre-Selecting Candidates: To cut down on computation, we did not pass every reference-candidate point pair through the LLM. Rather, for each reference point, we pre-select the 5 candidate points that have the closest cosine similarity to the reference point.

5.2.4 Ensuring 1-1 Matching

For the Cosine Similarity and LLM automated methods, we carry out a further step. Sometimes a single candidate point may be matched to multiple reference points. We further disambiguate the situation by finding the best-matching reference point for each candidate point that has more than one reference match. To do so, we developed the assignment algorithm described in Appendix A.5.

5.3 Calculating the Candidate Summary Recall Score

The weighted percentage of reference points that have at least one match, according to the automated method, is the candidate summary recall score given by the method. The weighting of reference points is described in Appendix A.3.

6 Results

6.1 Step 1: Point Extraction

To evaluate the performance of automated models for point extraction, we employed the *easiness* scores introduced in Zhang and Bansal, 2021 and further extended by Nawrath et al., 2024. This score assesses the similarity between generated points and human-written ones, and is described in Appendix A.6.

We compare our approaches with the following baselines:

Sentence split: A baseline approach where the list of generated points is simply the list of sentences from the given text.

A^{2}CU-Generator (Liu et al., 2023b): See section 2.2.

	E_R	E_P
Sentence split	.70	.81
A^{2}CU-Generator	.80	.72
Dense X	.85	.85
Fine-tuned Dense X	.88	**.91**
LLM with multi-shot prompt	**.91**	.89

Table 1: *Easiness* scores (Zhang and Bansal, 2021; Nawrath et al., 2024) for point extraction. E_R and E_P represent the recall- and precision-oriented *easiness* scores respectively.

Dense X (Chen et al., 2023): See section 5.1.1.

Performance

Table 1 presents the results for the point extraction task. Dense X is a strong baseline. Our fine-tuned model and our LLM approach achieve an improvement of 3-6 percentage points over this baseline.

A qualitative analysis of the extracted points is also instructive. As seen in the examples in Figure 2, the A^{2}CU-Generator, which was trained on non-legal data, produces points that are too granular and that do not capture the complex meaning of the original text. Some of the points are not proper propositions (they do not make a declarative statement). The fine-tuned Dense X model output is better but still contains inaccuracies. In particular, the last point in the example in the Figure is incorrectly stated as if it were a fact. In contrast, the LLM produces points that are correct and properly capture the original meaning.

6.2 Step 2: Point Comparison

We first evaluate the performance of automated methods for Step 2 using gold Step 1 points extracted by human annotators. We calculate two sets of performance metrics as follows.

Pairwise Matching Scores: Using the automated method, we predict whether each reference-candidate pair is a match, and compare this to the gold annotation. We then calculate a precision and recall score for the Step 2 method. This indicates how closely the method's predictions align with human labels, for every reference-candidate point pair. We call these the *pairwise matching* scores, and they are an indication of how well the automated method can distinguish whether two sentences are making the same point in the context of a court case. This calculation is done before the assignment algorithm described in section 5.2.4.

Reference Point Based Scores: After performing the assignment algorithm (for the Cosine and

Original Text

Relevant to the question of whether an order should be made and, if so, what order, will be a number of factors such as the financial benefit which the applicant has already received, or whether the applicant has failed to take advantage of a right under the foreign law to claim financial relief.

Points created by A^{2}CU-Generator

- Relevant to the question of whether an order should be made
- relevant to the question of what order should be made
- Relative to the question of whether an order should be made will be a number of factors
- Relative to the question of whether an order should be made and, if so, what order, will be a number of factors
- The financial benefit which the applicant has already received
- The financial benefit which the applicant has failed to take advantage of
- The financial benefit which the applicant has failed to take advantage of under the foreign law
- The applicant has failed to take advantage of a right
- The applicant has failed to take advantage of a right under the foreign law to claim financial relief

Points created by Fine-tuned Dense X

- Relevant to the question of whether an order should be made and, if so, what order, will be a number of factors.
- The financial benefit which the applicant has already received will be relevant factors.
- The applicant has failed to take advantage of a right under the foreign law to claim financial relief.

Points created by LLM

- A number of factors will be relevant to the question of whether an order should be made and, if so, what order.
- One relevant factor is the financial benefit which the applicant has already received.
- Another relevant factor is whether the applicant has failed to take advantage of a right under the foreign law to claim financial relief.

Figure 2: Examples of points produced by different automated Step 1 models. The A^{2}CU-Generator, which was trained on non-legal data, produces points that are too granular and that do not capture the complex meaning of the original text. Some of the points are not proper propositions (they do not make a declarative statement). The fine-tuned Dense X model output is better, but still not quite right – the last point, in particular, is incorrectly stated as if it were a fact. The LLM produces points that are correct and properly capture the original meaning.

LLM methods only), we then calculate another set of precision and recall scores for the method, from the frame of view of each reference point. Here, we are asking, for each reference point: if the automated method says there is a match, is there actually a match according to the gold annotation (and vice-versa)? This is regardless of which candidate point is matched. We call these scores the *reference point based* scores. These scores are an indication of how well the automated method can pick out which reference points are covered by the candidate summary. Since the summary recall score of the candidate summary is the percentage of reference points that are covered by the candidate summary, the *reference point based* scores also give an indication of how accurate the resulting summary recall score is likely to be.

Further details are given in Appendix A.7.

Performance

Table 2 shows the results of the automated methods for Step 2. Note that pairwise metrics were not calculated for the NLI-based method, because this method does not perform matching between two points, but rather asks if a candidate paragraph entails a reference point.

The F1 scores show that the LLM performs much better at this task than the other methods. This indicates that the LLM can better distinguish the nuances in complex legal statements than simpler models. The LLM many-shot and few-shot regimes perform similarly.

	Pairwise matching			Reference point based		
	P	R	F1	P	R	F1
Cosine similarity	.20	.67	.31	.62	.70	.66
A^{2}CU-NLI	n/a	n/a	n/a	.69	.49	.57
A^{2}CU-NLI $_{\text{fine-tuned}}$	n/a	n/a	n/a	.55	**.86**	.67
LLM, few-shot	.60	**.82**	**.70**	**.87**	.83	**.85**
LLM, many-shot	**.61**	.81	.69	**.87**	.84	**.85**

Table 2: Precision (P), recall (R) and F1-score of automated methods for Step 2. The *pairwise matching* scores are an indication of how well the method can distinguish whether two sentences are making the same point in a legal context. The *reference point based* scores indicate how well the method can pick out which reference points are covered by the candidate summary.

The absolute *pairwise matching* precision scores are not high. This indicates that, though it may seem a simple task to compare two sentences to see if they make the same point, this appears to be quite tricky for automated methods, even state-of-the-art LLMs that are given full context.

Because many of the false positives involve the same candidate point being matched to multiple reference points, the assignment algorithm in Section 5.2.4 mitigates the effect of these errors on the downstream summary recall score calculation, because each candidate point is only allowed to match to one reference point.

	Pearson Correlation			RMSE
	Summ.	Sys.	Pop.	
ROUGE-1	.350	.421	.523	.171
ROUGE-2	.651	.684	.595	.139
ROUGE-L	.656	.739	.676	.134
BERTScore	.596	.722	.589	.325
A^2CU	.830	.909	.638	.093
A^3CU	.477	.607	.048	.146
Pointwise $_{\text{DX-ft, NLI-ft}}$	.838	.883	.807	.236
Pointwise $_{\text{LLM, LLM-FS}}$	**.938**	**.987**	.940	.037
Pointwise $_{\text{LLM, LLM-MS}}$	.923	.975	**.950**	**.035**

Table 3: Pearson correlation (summary-, system- and population-level) of automated methods with human evaluation, as well as root mean squared error (RMSE) between automated metrics and human scores.

6.3 Comparison with Human Evaluation

We run Step 1 and Step 2 in a fully automated manner, obtaining recall scores for each candidate summary. We then calculate the correlation of these automatically-calculated recall scores with the recall scores obtained from human annotation (described in Section 4).

Due to computational resource limitations, we focused on only these combinations of Step 1 and Step 2 methods:

Pointwise $_{\text{DX-ft, NLI-ft}}$ is the non-LLM version, using the fine-tuned Dense X model for Step 1 and fine-tuned A^2CU-NLI model for Step 2.

Pointwise $_{\text{LLM, LLM-FS}}$ uses the LLM for Step 1 and few-shot-prompted LLM for Step 2.

Pointwise $_{\text{LLM, LLM-MS}}$ uses the LLM for Step 1 and many-shot-prompted LLM for Step 2.

We calculate three types of correlation scores. The summary-level score is the average (over all m cases) of the correlation across the n candidate summaries for each case. The system-level score first averages (over all m cases) the scores of the candidate summaries for each system (i.e. LLM and prompt that generated the summary), then calculates the correlations across the n systems using these average scores. In addition, we calculate a population-level correlation score, where the $m \times n$ candidate summaries are each considered as an individual datapoint in the correlation.

The correlation results are shown in Table 3. The LLM-based pointwise methods produce higher correlations (for all three correlation types) than the baselines. The non-LLM-based Pointwise $_{\text{DX-ft, NLI-ft}}$ performs better in some of the correlation categories than the baselines, but not as well as the LLM-based pointwise methods. This shows that the use of advanced LLM models yields a significant advantage in this task involving complex legal text.

We calculate the significance (p-value) of the improvement in correlation of our best-performing method over the best baseline, using the PERM-BOTH permutation algorithm described in Deutsch et al., 2021. Because the summary- and system-level correlations involve averaging over the cases, each correlation is calculated over only 5 systems, which is too small a number to achieve significance. For the population-level correlation, however, our method shows a strongly statistically significant improvement ($p < 0.001$) over the best baseline.

In addition, the root mean squared error between the LLM-based pointwise metric and the human metric is less than half that of the best baseline.

Figure 3 plots the summary recall scores obtained from several automated metrics against the human scores. We see that the pointwise metric corresponds much more closely with human evaluation than the baselines do. The pointwise metric has a narrower spread, and a best-fit line much closer to the ideal line, than the baselines.

7 Discussion and Conclusions

The improvement in correlation of our method over the baselines is particularly pronounced for the population-level correlation. This is an indication that our method produces consistent results across all the court cases in our dataset. In other words, it does not merely rank the candidate summaries for each case in the correct order from best to worst, but also gives a recall score that is well-correlated with the human score on an individual candidate summary level.

In addition, the root mean squared error between the LLM-based pointwise metric and the human metric is much smaller than that of the baselines. This indicates that our method produces absolute recall scores that are close to the human scores, thus giving an accurate idea of the absolute quality of a single LLM summary (and not just the comparatative quality of multiple LLM summaries).

The plots in Figure 3 illustrate these points further.

Apart from performance, one of the advantages of the pointwise evaluation method over existing ones its its interpretability and explainability. The method allows us to see exactly which reference points are included or missing in the candidate summary. This allows us to improve the candidate

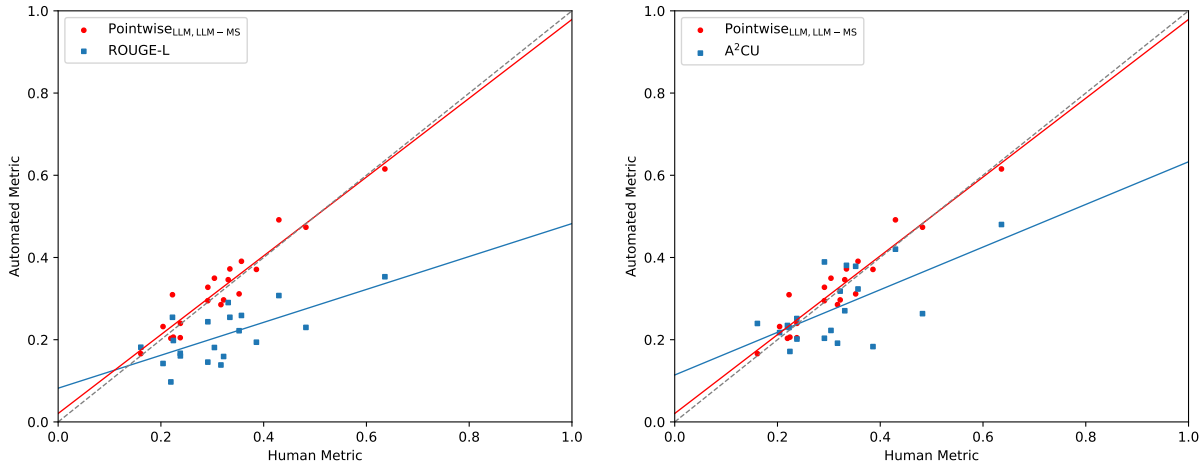

Figure 3: Correspondence of automated metrics with human evaluation. These plots show one of the LLM-based pointwise metrics (red circles), as well as the two best baselines, A^{2}CU and ROUGE-L (blue squares). Each individual point in the scatterplot represents one generated candidate summary. The best-fit lines are also shown. The gray dashed line represents a perfect match. The pointwise metric has a narrower spread, and a best-fit line much closer to the ideal gray dashed line, than the baselines.

summaries in a targeted manner, for example by editing the prompts to tell the summarizing LLM to focus more on the type of information that the current summaries do not include. We can also see which points in each candidate summary were not included in the reference (and thus are probably irrelevant), and can thus also improve the LLM prompts to avoid these.

8 Limitations and Further Work

The pointwise evaluation methodology focuses on the content of a summary, and does not account for (more subjective) aspects of a text, such as writing style and flow. These aspects are nevertheless an important part of a well-written legal summary.

Creating a meta-evaluation dataset for long-form legal summaries is very resource-intensive, and we were thus only able to create a small dataset. Future work to extend the dataset to more cases and across more jurisdictions would allow for more representative and statistically significant tests.

The pointwise method currently compares candidate summaries to a single human-written summary. Using multiple human-written references, as done in the original Pyramid (Nenkova and Passonneau, 2004) and LitePyramid methods (Shapira et al., 2019), could improve the robustness of the method.

It would also be instructive to explore the use of other LLMs (other than Claude 3.5 Sonnet) for the Step 2 task of determining whether two points are saying the same thing in a legal context.

The greedy assignment method described in section 5.2.4 may not always assign a candidate point to the correct reference point. A more complex, non-greedy algorithm may improve the matching and be closer to how a human would pick the best pairwise matches between two sets of points.

Because of the complexities involved in legal reasoning, perfect one-to-one matches between points may not always be possible; this could be an interesting direction for future work.

The pointwise method is more computationally-intensive than baselines such as ROUGE, but the computations can be parallelised for greater efficiency.

We developed the pointwise evaluation methodology for the specific task of evaluating legal summaries. It is appropriate for use cases where there is an objective standard for the content that should or should not be included in a text. It would be less appropriate for use cases where there are many possible interpretations of a topic, such as arguing for or against a particular issue.

9 Ethics

The impacts – and potential harms – of artificial intelligence are ever-increasing, and sensitive domains like legal technology can often experience outsized effects from misuse. Over the course of the research performed, we sought to ensure that any data and results – generated or derived – were free of such harms. Our work was built upon court opinions and judgments that reference real parties,

locations, and accusations, though we took care to ensure this information remained neutral and without commentary during the model development processes. Further, we took steps to ensure that no individual, entity, or party was unfairly targeted or identified, opting to leverage very high visibility cases drawn from the UK Supreme Court.

Despite managing all items under the scope of our control in the manner described above, the work and experimentation performed under this research effort does leverage pretrained large language models for tasks such as data augmentation, passage extraction, and pointwise comparison (among others). Such models are generally built and hosted by third parties, and may hold inherent biases, shortcomings, or factual inconsistencies based on the processes and data with which they were trained. These potential limitations were not *exhaustively* studied under the work contained in this paper, though we reviewed the results to the best of our determinative ability to ensure they met these ethical standards.

Nonetheless, we implore researchers who wish to leverage this work to likewise verify that potential hallucinations are limited, biases are minimized, and model-based decision making is fair and explainable. We discourage leveraging this work for critical decision making in any legal, personal, or high-risk domain without thorough review of results by a trained subject-matter expert (e.g., a licensed attorney specializing in the area of interest). Further, we invite future researchers to ensure that similarly appropriate disclosures are made to any end users consuming data or insights drawn from this work.

References

Rishabh Agarwal, Avi Singh, Lei M. Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, John D. Co-Reyes, Eric Chu, Feryal Behbahani, Aleksandra Faust, and Hugo Larochelle. 2024. Many-shot in-context learning. *Preprint*, arXiv:2404.11018.

Ansar Aynetdinov and Alan Akbik. 2024. Semscore: Automated evaluation of instruction-tuned llms based on semantic textual similarity. *Preprint*, arXiv:2401.17072.

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.

Tong Chen, Hongwei Wang, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. 2023. Dense x retrieval: What retrieval granularity should we use? *Preprint*, arXiv:2312.06648.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Hoa Trang Dang and Karolina Owczarzak. 2008. Overview of the TAC 2008 update summarization task. In *Proceedings of the First Text Analysis Conference, TAC 2008, Gaithersburg, Maryland, USA, November 17-19, 2008*. NIST.

Daniel Deutsch, Rotem Dror, and Dan Roth. 2021. A statistical analysis of summarization evaluation metrics using resampling methods. *Transactions of the Association for Computational Linguistics*, 9:1132–1146.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ben Hachey and Claire Grover. 2006. Extractive summarisation of legal texts. *Artificial Intelligence and Law*, 14(4):305–345.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: decoding-enhanced bert with disentangled attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Huan Yee Koh, Jiaxin Ju, Ming Liu, and Shirui Pan. 2022. An empirical survey on long document summarization: Datasets, models, and metrics. *ACM Comput. Surv.*, 55(8).

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yixin Liu, Alex Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023a. Revisiting the gold standard: Grounding summarization evaluation with robust human evaluation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4140–4170, Toronto, Canada. Association for Computational Linguistics.

Yixin Liu, Alexander Fabbri, Yilun Zhao, Pengfei Liu, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023b. Towards interpretable and efficient automatic reference-based summarization evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16360–16368, Singapore. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Marcel Nawrath, Agnieszka Nowak, Tristan Ratz, Danilo Walenta, Juri Opitz, Leonardo Ribeiro, João Sedoc, Daniel Deutsch, Simon Mille, Yixin Liu, Sebastian Gehrmann, Lining Zhang, Saad Mahamood, Miruna Clinciu, Khyathi Chandu, and Yufang Hou. 2024. On the role of summary content units in text summarization evaluation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 272–281, Mexico City, Mexico. 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.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Sentence Transformers. 2024. Pretrained models. `https://sbert.net/docs/sentence_transformer/pretrained_models.html`.

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.

Abhay Shukla, Paheli Bhattacharya, Soham Poddar, Rajdeep Mukherjee, Kripabandhu Ghosh, Pawan Goyal, and Saptarshi Ghosh. 2022. Legal case document summarization: Extractive and abstractive methods and their evaluation. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1048–1064, Online only. Association for Computational Linguistics.

The Supreme Court of the United Kingdom. 2024. Decided cases. `https://www.supremecourt.uk/decided-cases/index.html`.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. 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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International*

Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net.

A Appendix

A.1 Step 1: Point Extraction

In breaking down the summary into discrete points, we use the following heuristics based on the nature of legal text.

- If a sentence specifies who said something (or which law specifies something), this is included in the resulting point(s).
- If there is a list of multiple factors that are considered, each factor is split into a separate point.
- A sentence that follows a *because-therefore* structure can be split into separate points.
- A sentence that has a conditional structure and cannot be split without changing the meaning of the sentence should be left as a single point, even if the resulting point is quite long.

Examples are shown in Figure 4.

A.2 Step 2: Point Matching

For each reference point, we find the best-matching point in the candidate summary (if any).

A one-to-one match is done where possible – i.e. each reference point should be matched to at most one candidate point, and vice-versa. If one or more candidate points each only cover part of the reference point's content, we mark these as "partial matches".

Sometimes multiple candidate points are sufficiently similar to the reference point to be considered a match. In such cases the best match is annotated as a full match, and the rest are noted as "other relevant matches". Sometimes there may be multiple reference points talking about the same thing – such as where the court, in its reasoning, repeated a point already stated in the case background for emphasis. If these multiple similar reference points may match to a single candidate point, the reference point with the most similar context to the candidate point is marked as the "full match", and the remaining reference points get the candidate point as an "other relevant match".

A.3 Weighting Scheme for Recall Score

A single sentence in the original summary may be broken down into multiple reference points, with many shared words between the points. This is particularly the case when there are multiple factors mentioned in the original sentence (see the first example in Figure 4). In such cases, the resulting points would have an oversized effect on the final recall score of the document.

To mitigate this problem, we apply the following weighting scheme to the reference points, where the weight W_p of each reference point p (containing lemmas each denoted with l) is:

$$W_p = \frac{\sum\limits_{l \subset p} W_l}{\sum\limits_{p} \sum\limits_{l \subset p} W_l},$$

where

$$W_l = \begin{cases} \min(\frac{N_{l,\text{ para}}}{N_{l,\text{ points}}}, 1) & \text{if lemma in paragraph} \\ 0 & \text{otherwise.} \end{cases}$$

Here, $N_{l,\text{ para}}$ is the number of times the lemma appears in the original paragraph, and $N_{l,\text{ points}}$ is the total number of times the lemma appears in all the points extracted from that paragraph. This weighting scheme down-weights points which "share" many lemmas with other points, where these lemmas did not appear as often in the original paragraph.

In addition, we also consider the type of match (full or partial): a reference point with a full-matching candidate point will count fully towards the recall score. A reference point that has no full match but one or more partial matches has its contribution reduced by a factor of 0.5.

A.4 Explanation Provided to Step 2 LLM

The following text is included in the prompt for the LLM for Step 2 (point matching), to specify what "making the same point" means in the context of a legal summary:

```
Two sentences make the same point if they
explain the same legal reasoning step,
describe the same part of a legal test or
rule, describe the same conclusion by the
same court, or give the same background
information about the facts and events
about a case.
Note in particular the following
situations when two sentences do NOT make
the same point:
If the sentences seem to be making the
same argument, but the argument is being
```

Original Text

The court agreed that the scale of the publications, the plaintiff's situation, and the gravity of the statements themselves supported the finding of serious harm.

Points

- The court agreed that the scale of the publications supported the finding of serious harm.
- The court agreed that the plaintiff's situation supported the finding of serious harm.
- The court agreed that the gravity of the statements themselves supported the finding of serious harm.

Original Text

There were uncertainties surrounding the underlying facts of the case, making it difficult to ascertain the precise scope of the doctrine.

Points

- There were uncertainties surrounding the underlying facts of the case.
- These uncertainties made it difficult to ascertain the precise scope of the doctrine.

Original Text

Section 103A provides that a dismissal is unfair if the reason for the dismissal is that the employee made a protected disclosure.

Points

- Section 103A provides that a dismissal is unfair if the reason for the dismissal is that the employee made a protected disclosure.

Figure 4: Examples of how complex legal texts are split into points.

`made by different parties (e.g. the court and the plaintiff), the sentences are not considered to be making the same point.`
`If the sentences seem to be making the same argument, but the argument is being made by different courts (e.g. the Supreme Court and the Court of Appeal), the sentences are not considered to be making the same point.`
`If the sentences are describing two different parts of the same legal test or rule, they are not making the same point.`
`If one sentence talks about a conclusion and one sentence focuses on the reasoning behind the conclusion, they are not making the same point.`

A.5 Assignment Algorithm to Ensure 1-1 Matching

To find the best-matching reference point for each candidate point that has more than one reference match, we developed the following greedy algorithm (combined with a further prompt to an LLM in the LLM case).

Let the set of candidate points that have at least one reference match be C. The set of reference points that the candidate points in C match to is R. For each candidate point c_m in C, if c_m has only one reference match r_i, this reference is assigned to c_m. r_i and c_m are then removed from the pools R and C. This algorithm is run recursively until there are no more candidate points in C that have only one reference match.

We then sort the remaining candidate points in C in increasing order of the number of reference points they each match to. We then find the best match for each candidate point c_n in C thus:

- For the cosine case, we assign to c_n the reference point with the smallest cosine distance from c_n. We then run the algorithm described in the previous paragraph again.
- For the LLM method, we use a further prompt to an LLM. We run the candidate point c_n through an LLM prompt, together with all the reference points it matches to, and ask the LLM which of the reference points is the closest match. The LLM is prompted with instructions for what is and is not considered a similar point in the context of a legal case. The LLM's answer r_j is assigned to c_n, and r_j and c_n are removed from the pools R and C. After each LLM call (which makes one assignment of an r to a c), we then run the previously-described algorithm again.

We proceed in this way until all candidate points which had multiple reference matches have been assigned a single reference point.

A.6 Easiness Score Calculation for Point Extraction

The *easiness* score (Zhang and Bansal, 2021; Nawrath et al., 2024) is composed of a recall-based and a precision-based metric computed between human-labeled points (P^H) and generated points (P^G). The recall-oriented metric (E_R) measures whether for each human-written point, there is a closely matching generated point. The precision-oriented score (E_P) measures whether for each generated point, there is a closely matching human-written counterpart.

For a given passage with M human-written points and N generated points, these scores are defined as follows:

$$E_P = \frac{\sum Acc_j}{N},$$

where

$$Acc_j = \max_m Rouge1_{F1}(P_j^G, P_m^H).$$

The recall-based score is then computed in the reverse direction:

$$E_R = \frac{\sum Acc_j}{M},$$

where

$$Acc_j = \max_n Rouge1_{F1}(P_j^H, P_n^G).$$

The ROUGE score is used here (rather than, for example, embedding similarity) because we expect point extraction (which more closely resembles a chunking process than a paraphrasing one) to preserve the original lemmas for the most part.

A.7 Step 2 Performance Metrics

For each reference point, we pre-select the 5 candidate points that have the closest cosine similarity to the reference point. This forms the pre-filtered set of reference-candidate pairs for which we will calculate a precision and recall score for the method. For the purposes of calculating pure Step 2 performance of the LLM method, if there are gold matches that do not make it into the top 5 candidate points, we include these pairs as well. This allows us to calculate the real performance of the LLM method even if the cosine method produces a false negative.

We compare the automated method predictions to the gold labels as follows to calculate the pairwise matching score. Where the gold annotation indicates a "full match" or "other relevant match" and the automated method indicates a match, count this as a True Positive. Where the gold annotation indicates a "partial match" and the automated method indicates a match, count this as half a True Positive. Where the human annotation indicates no match at all, and the method indicates a match, count this as a False Positive. Where the human annotation indicates a "full match" or "other relevant match", and the automated method does not indicate a match, this is a False Negative. Where the gold annotation indicates a "partial match" and the automated method does not indicate a match, this is half a False Negative. All other cases are True Negatives.

A.8 LLM Prompts

The LLM prompts for proposition extraction, proposition comparison, and the assignment algorithm are available upon request.