Abstracting Terms of Service

By Nicholas Lin, Sophie Chance, Yoni Nackash

Abstract

Appearance of Terms of Service (ToS) in everyday life is continuing to increase as humans become more dependent on digital products and services. These documents are often long and difficult to understand, and it is common for individuals to accept terms of service blindly. This paper proposes to classify and summarize ToS documents to make these agreements more understandable. Using a human-labeled dataset from Terms of Service; Didn't Read (ToS;DR), the system uses a two-stage approach: first, a BERT-based classification model identifies important clauses within the ToS documents at the chunk level; second, a paraphrasing pipeline feeds the data from stage one into fine-tuned T5 and GPT-4o-mini models, transforming these sentences into simplified summaries. The results demonstrate that the classification model achieves high recall of 0.826, minimizing the risk of missing critical information, and the paraphrasing stage produces accessible summaries optimized for clarity and relevance with a BERTScore of 0.873.

Introduction

In today's digital world, individuals are required to agree to ToS agreements for nearly every online service they use. These documents often contain crucial information about user rights, data privacy, and legal obligations. However, their complexity and length make them inaccessible to most people. As a result, users routinely accept these agreements without a full understanding of what they are agreeing to, leaving them vulnerable to exploitation.

This paper seeks to address this problem by leveraging Natural Language Processing (NLP) to analyze ToS documents and identify their most important points. Using human-labeled data from ToS;DR, models are trained to extract and summarize critical points from these agreements, presenting the information in a way that is clear and concise. The goal is to create a tool that ensures users are aware of the key terms they are agreeing to, helping to restore transparency between service providers and users.

Background

The complexity of ToS agreements has led to increasing efforts to use NLP to improve user comprehension. Existing research performed by Manor et al. (2019) demonstrated a reasonable ability to summarize privacy-related clauses in ToS documents. Their work emphasized the importance of addressing the unique challenges of verbose legal texts and the need for concise, user-friendly summaries. Qualitatively, they saw higher success rates when the reference texts they were summarizing were shorter. Building on their efforts, this paper's approach integrates a hybrid pipeline that includes classification before abstractive summarization, in order to first identify the most important information to summarize in ToS agreements, and then distill them. This research contributes to the application of NLP in legal texts by bridging the gap between clause identification and accessible summarization.

Methods

To implement an effective system, the process was divided into the following components: data collection and preparation, initial classification, and paraphrasing.

Data Collection and Preparation

Data was gathered from ToS;DR, an open platform that provides human-labeled highlights from ToS agreements for hundreds of companies, such as Amazon, Telegram, and GrubHub. Data was retrieved using the ToS;DR API and supplemented with web scraping to ensure a comprehensive dataset.

In total, the dataset included ToS documents from 171 companies. Each company's ToS was broken out into several documents, such as Privacy Policy and FAQs. Within each document, the dataset included sentences that have been marked as important, as well as paraphrased summaries of these highlighted sentences. These highlights and paraphrases served as the foundation for training and evaluating the two-stage system: the highlighted sentences were used for training classification models with a goal of identifying text as important or not, while the paraphrases of highlighted sentences were used for training abstractive summarization models.

To clean the data, highlights with significant outlying lengths were removed, as these were counterintuitive to the goal of creating accessible text. Regex cleaning was performed to remove non-english characters, non-essential punctuation, and HTML related tags from web-scraping. However, due to time constraints and the nature of web-scraped data there were still portions of data with irregular text.

Due to transformers having a limited max sequence length for inputs, the decision was made to experiment with the data processed in two ways: a sentence-level and a chunk-level breakdown. For both, NLTK's sentence tokenizer was used to break down sentences of entire ToS documents. Ultimately, the dataset contained 50,807 rows of sentences, of which 3,617 rows had highlights.

Sentences without provided highlights from the data were labeled as non-important (0) and sentences with highlights were labeled as important (1) (Appendix A Table 1.1). For the chunk-level breakdown, the ToS sentences were grouped into chunks of 3 sentences, where each chunk had a one sentence overlap with the previous chunk. Similarly to the sentence breakdown, chunks without a highlighted sentence were labeled as non-important (0) and chunks that included a highlighted sentence were labeled as important (1) (Appendix A Table 1.2).

For both sentence and chunk-level breakdowns, the data was split into training and testing sets using an 80/20 train-test split. The classes in the training were balanced to avoid overfitting on the majority class.

Stage 1: Classification

To identify key highlights in ToS documents, several models were developed to classify text importance for both sentence and chunk-level text. The full results can be seen in Appendix A Table 2.1.

Baselines: For the initial baseline before class balancing, classifying all data as the majority label yielded 80% accuracy. For a second level baseline, scikitlearn's TF-IDF vectorizer was used to transform the training and test data. The vectorized data was then used to train a simple logistic regression model. Passing in the testing dataset, the results were promising for both the sentence and chunk-level data, however both showed signs of overfitting.

CNN: In an attempt to better capture the patterns within the text, the next experiment included a Convolutional Neural Network (CNN). Again, the model seemed to overfit on both the sentence and chunk-level data. There was no significant improvement on the testing accuracy when compared to the baseline. Looking through the test predictions, it became apparent that the model was getting confused on

a large variety of samples. This could be because CNNs are better at capturing local patterns, rather than being able to understand the full context of a sentence.

T5: The next experiment used the pretrained T5 for Sequence Classification model. Since the T5 model was trained on a text-to-text transfer learning approach, it might be better suited to deal with diverse sentence inputs from the web-scraped data. Due to the nature of web-scraping, there was always the possibility of noisy data. Various hyperparameter tunings were attempted, such as adjusting learning rate and input prefixes. For the sentence-level data, the T5 was performing significantly worse than the CNN. The T5 was able to learn incrementally more with the chunk-level data, but overall there was no significant improvement using the T5 over other models.

BERT: To continue strengthening context understanding within the text, the next experiment employed attention through the use of BERT. The CLS token was used as an input to a neural network classification layer, with the goal of capturing the contextual meaning of the entire input. At this point, hyperparameter tuning with the sentence-level and chunk-level data yielded different results. Freezing certain layers of the BERT model and reducing the hidden layer size of the neural network architecture helped to reduce overfitting in the sentence-level data. Even with hyperparameter tuning, the sentence-level data was not performing great with its limited context of sentence importance.

When tokenizing the chunk-level data, separator tokens were added between the sentences within each chunk, as demonstrated by Lui (2019) in the original BERTSUM paper. The intention here was for the model to be able to recognize that each sentence within a chunk could have its own context. The CLS token was then used as an input to a classification layer. Through experimentation, unfreezing the BERT embedding weights, and freezing the first 6 BERT encoding layers resulted in the least amount of overfitting. Overall, training on chunk-level data with the fine-tuned BERT model achieved a slight improvement over the baseline. For the complete results, refer to Table 2.1 in Appendix A.

The various experiments showed that the classification models generally performed better using the chunk-level data. This seems reasonable, as inputs with more sentences have more contextual representation. The decision was made to select the BERT classification model, using chunk-level data, as the Stage 1 classifier.

Identifying Sentence Contribution

Moving forward with the chunk-level data introduced one additional hurdle: identifying which sentence within an “important” chunk was the true highlight. In other words, which sentence had the highest contribution to the chunk’s classification of “important”? Several experiments were attempted to inform this sentence contribution. To start, chunks that were classified as “not important” were filtered out of the results – only “important” text will eventually be passed on to the Stage 2 abstractive paraphrasing, so sentence contribution analysis was only needed for “important” chunks.

The first experiment extracted the BERT attention weights for each sentence within a chunk, and compared them to see which sentence contained the highest attention weights. Pooling was needed to calculate the overall attention for each sentence. Whether pooling by sum, mean, or max, this experiment did not prove successful – the correct sentence was identified roughly 30% of the time.

The next experiment leveraged cosine similarity to compare the CLS token with the token embeddings of each sentence within a chunk. The goal was to determine which sentence vector was most similar to the CLS token. Pooling was again needed, as embeddings are at the token level. This experiment yielded very similar results to the attention weight experiment, with the correct sentence being

identified roughly 33% of the time. Taking a deeper look at examples that consistently identified the wrong sentence, it became apparent that highlights are often surrounded by other important sentences. This seems reasonable, as it often takes more than one sentence to make an important point. In addition, the sentence contribution experiments only looked at weights and embeddings from the fine-tuned BERT model, which may not be telling the entire story. As explained above, the classification model takes the outputs from BERT and passes them to a series of dense layers. The sentence contribution experiments are not taking these additional layers, and what they may have learned about the text, into account when identifying which sentence it thinks is the most important.

Table 1 - Incorrect Sentence Contribution Example

We retain the right to create limits on our services at any time with or without notice. We may also impose limits on services or aspects of them or restrict your access to part or all of our services without notice or liability. We may change, suspend, or discontinue any or all of our services at any time, including availability of any product, feature, database, and content yours or ours.

True highlight

Highlight identified via sentence contribution

It became clear that these experiments would not be sufficient in determining which sentence from each chunk should be passed forward Stage 2. To remedy this, a solution was built to count the occurrences of sentences within "important" chunks. Due to the overlapping nature of chunks introduced during data preparation, highlight sentences within "important" chunks will appear multiple times. The idea here is the more a sentence appears in the dataset, and if the sentence is in a chunk classified as "important", the more important the sentence itself is. Thus, for each chunk, the sentences with the highest occurrence count were passed on to Stage 2 abstractive paraphrasing.

Stage 2: Paraphrasing

The highlights identified in Stage 1 were passed through a paraphrasing pipeline to generate concise summaries. This step was essential to make the extracted information more accessible for end users.

Baseline: The initial baseline evaluated the accuracy of the T5 model without any additional training or fine-tuning. The T5 model was chosen for its ability to handle text-to-text transformations, in this case, paraphrasing. To assess its effectiveness, the test set from the ToS;DR dataset was used to evaluate the highlights and their corresponding paraphrases.

Each highlight was passed into the T5 large model with the prompt "paraphrase: {highlight}". This enabled the evaluation of the T5 model on ToS text without the domain-specific training. However, the outputs clearly were not high-quality. Paraphrases often lacked relevance or failed to simplify the text effectively, revealing the need for additional training.

T5: Early experimentation with the T5 model showed that even minimal fine-tuning significantly improved its ability to generate effective paraphrases. The model was fine-tuned on highlights from the ToS;DR dataset. Additionally, the model's performance was further refined by incorporating document-specific context into the prompts. The full prompt used was: "Paraphrase the following {doc_type} text while maintaining the same meaning and staying very concise:\n\n{highlight}". In this prompt, {doc_type} refers to the section type (e.g., "Cookie Policy" or "Privacy Statement"), and {highlight} refers to the specific highlighted text.

GPT-4o-mini Waterfall: To ensure high-quality outputs while maintaining cost efficiency, a waterfall approach was implemented using BERT and SpaCy scores as quality metrics. This approach addresses cases where the generated paraphrase may lack semantic relevance or accuracy compared to the original highlight. A T5 paraphrase was flagged as inaccurate if the BERT score fell below 0.75 or the SpaCy score fell below 0.5. In these cases, the highlight was passed to a fine-tuned GPT-4o-mini model for paraphrasing. The outputs from both models were then evaluated using BERT and SpaCy scores, with the highest-scoring paraphrase selected.

By integrating this two-model process, the waterfall approach minimizes the use of the more expensive GPT model. The T5 model handles the majority of paraphrasing tasks efficiently, and the GPT-4o-mini model is invoked only for low-quality cases, optimizing both performance and cost.

Results & Discussion

Model Orchestration

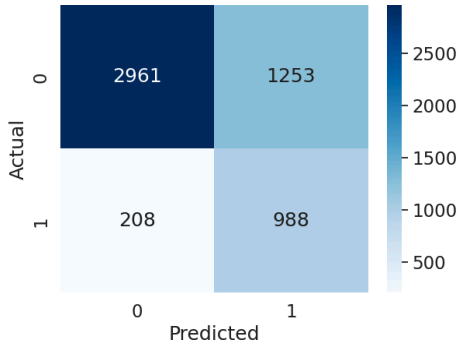
The final step brought all the components together in a full orchestration pipeline. First, the test ToS dataset was passed through the best performing BERT classification model that was trained in Stage 1, and chunks were classified. Next, the important chunks flowed through the sentence contribution scorer, to identify the most significant sentences within each chunk. Lastly, these sentences were then fed into the trained Stage 2 T5/GPT-4o-mini model for paraphrasing.

Classification Results

Table 2 below shows the results from the Stage 1 BERT classification. The model was able to optimize for a considerably low number of false negatives, and therefore a high recall score. This is optimal for the problem statement of classifying important text from legal documents, as it is preferable to over highlight unnecessary text (false positives), rather than miss true important text (false negatives).

Table 2 - Stage 1 Evaluation Metrics and Confusion Matrix of BERT classification

Metric	Value
Accuracy	0.729
Recall	0.826
Precision	0.441
F1 Score	0.575



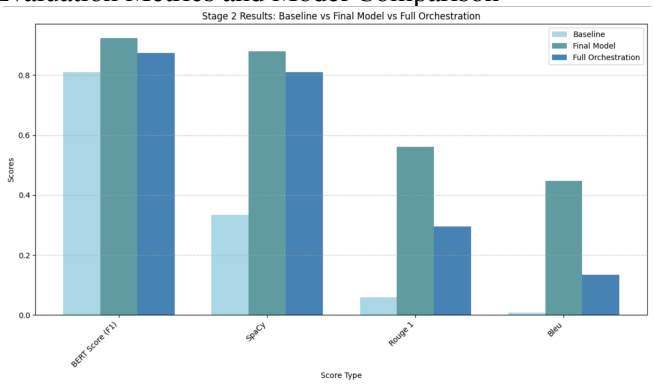
Paraphrasing Results

The baseline T5 model's low scores (Appendix B Table 1.1) confirm that the model struggles to generate high-quality paraphrases without domain-specific fine-tuning. While the BERT Score (F1) is relatively high at 0.810, other metrics are revealing. The SpaCy score of 0.334 highlights the lack of semantic similarity, while the low Rouge 1 (0.059), Rouge 2 (0.012), Rouge L (0.048), and Bleu (0.009) scores indicate that the model fails to generate accurate or simplified paraphrases. These scores are confirmed by reading the results which are often copied directly or return an irrelevant response such as "True".

Fine-tuning the model significantly improved all metrics. The BERTScore increased to 0.924, and the SpaCy score rose dramatically to 0.880, showing that the model produced more semantically relevant paraphrases. Additionally, Rouge 1 improved to 0.560, Rouge 2 to 0.465, Rouge L to 0.545, and Bleu to 0.447. These improvements demonstrate that fine-tuning enabled the model to better capture the nuances of legal language and generate concise, relevant paraphrases.

Figure 3 - Stage 2 Evaluation Metrics and Model Comparison

Score Type	**Baseline**	**Final Model**	**Full Orchestration** (True Positives)
BERT Score (F1)	0.810	0.924	0.873
SpaCy	0.334	0.880	0.810
Rouge 1	0.059	0.560	0.296
Bleu	0.009	0.447	0.134



In the full orchestration, which only includes the true positive results, the BERTScore decreased slightly to 0.873, and the SpaCy score dropped to 0.810. The Rouge 1 (0.296), Rouge 2 (0.159), and Rouge L (0.273) scores also saw reductions compared to the fine-tuned model. However, these scores are still significantly higher than the baseline. The drop in scores is expected because, in Stage 2, the model was provided with highlights generated by Stage 1, which in some cases differed significantly from the original input, making it more challenging to produce accurate paraphrases. Additional insights into the bias of the model based on the document type can be found in Section 2 of Appendix B.

Together, the two models performed well, outputting a list of reasonable take-aways (Appendix B Table 1.3) that could be used to decipher a complicated legal document. However, there is room for improvement. Examples of common mistakes can be found in Table 1.1 in Appendix C. Primarily, future iterations could be improved by better data cleaning, more training, and refining the workflows to increase consistency. For more details on this, reference Appendix C - Next Steps.

Conclusion

This paper works to address the complexity of ToS agreements by developing a two-stage system to identify and simplify critical information from hard-to-read documents. Using a combination of classification and text generation models, this approach effectively highlights important phrases before summarizing them. As compared to Manor et al. (2019), this paper achieved higher success rates due to the fact that the initial classification provided more focused text to be summarized. In addition, the inclusion of BERTScore and SpaCy metrics provide semantic evaluation of the generated paraphrases, rather than relying solely on token level evaluation with Rouge, as Manor does.

The two-stage approach outlined in this paper demonstrates the potential of NLP to make complex legal documents more understandable and equitable. By simplifying ToS agreements, this system empowers users and helps protect them from unknowingly agreeing to unfair terms.

Citations

Liu, Y. (2019). Fine-tune BERT for extractive summarization. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1903.10318

Lukose, E., De, S., & Johnson, J. (2022). Privacy pitfalls of online service terms and conditions: A hybrid approach for classification and summarization. In N. Aletras, I. Chalkidis, L. Barrett, C. G. Cătălina, & D. Preoțiuc-Pietro (Eds.), Proceedings of the Natural Legal Language Processing Workshop 2022 (pp. 65–75). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.nllp-1.6

Manor, L., & Li, J. J. (2019). *Plain English Summarization of Contracts*. In Proceedings of the 1st Workshop on NLP for Internet Freedom (NLP4IF): Censorship, Disinformation, and Propaganda (pp. 1-7). Association for Computational Linguistics. https://aclanthology.org/W19-2201

Palivela, H. (2021). Optimization of paraphrase generation and identification using language models in natural language processing. International Journal of Information Management Data Insights, 1(2), 100025. https://doi.org/10.1016/j.jjimei.2021.100025

Sun, Xiaobing & Lu, Wei (2020). Understanding Attention for Text Classification. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 3418–3428. https://aclanthology.org/2020.acl-main.312.pdf

“Terms of Service; Didn’t Read.” Frontpage, tosdr.org/. Accessed 4 Dec. 2024.

Appendix

Appendix A - Stage 1

Section 1 - Methods

Table 1.1 - Processed Sentence-Level Sample Dataset

Sentence	Highlight	Label
Telegram is a messaging app with a focus on speed and security, its superfast, simple and free.	NA	0
We do not use cookies for profiling or advertising.	We do not use cookies for profiling or advertising.	1

Table 1.2 - Processed Chunk-Level Sample Dataset

Chunk	Highlight	Label
Telegram is a messaging app with a focus on speed and security, its superfast, simple and free. You can use Telegram on all your devices at the same time your messages sync seamlessly across any number of your phones, tablets or computers. With Telegram, you can send messages, photos, videos and files of any type doc, zip, mp3, etc, as well as create groups for up to 200,000 people or channels for broadcasting to unlimited audiences.	NA	0
The only cookies we use are those to operate and provide our Services on the web. We do not use cookies for profiling or advertising. The cookies we use are small text files that allow us to provide and customize our Services, and in doing so provide you with an enhanced user experience.	We do not use cookies for profiling or advertising.	1

Section 2 - Classification

Table 2.1 - Classification Model Performances

Model	Train Accuracy: Sentence	Test Accuracy: Sentence	Train Accuracy: Chunk	Test Accuracy: Chunk
Baseline - Logistic Regression	0.805	0.646	0.822	0.704
CNN	0.768	0.641	0.820	0.727
T5	0.570	0.306	0.545	0.490
BERT Classification	0.553	0.333	0.868	0.729

Section 3 - Classification Results

Figure 3.1 - Stage 1 Final BERT Classification Model Confusion Matrix on Test Data

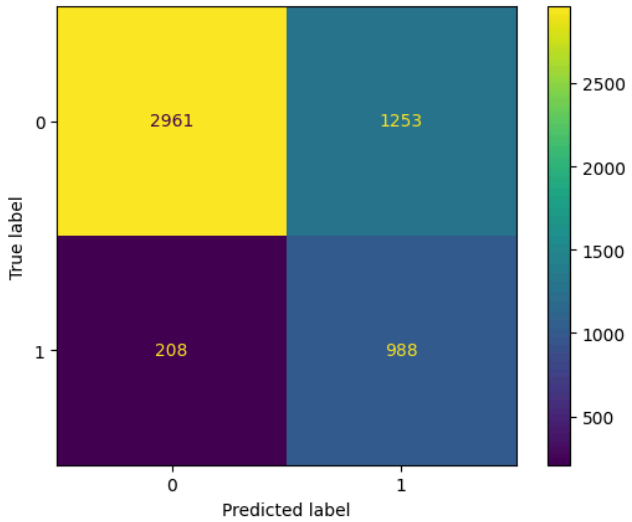
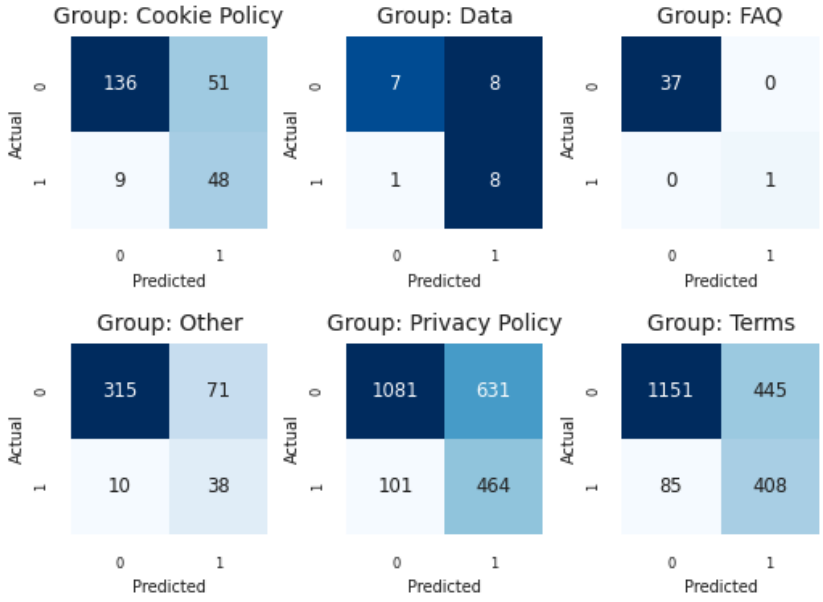


Figure 3.2 - Stage 1 Confusion matrix by segment type



Appendix B - Stage 2

Section 1 - Results & Discussion

Table 1.1 - Stage 2 Full Results

Score Type	**Baseline**	**Final Model**	**Full Orchestration** (limited to only those with a given paraphrase)
BERTScore (F1)	0.810	0.924	0.873
SpaCy	0.334	0.880	0.810
Rouge 1	0.059	0.560	0.296
Rouge 2	0.012	0.465	0.159
Rouge L	0.048	0.545	0.273
Bleu	0.009	0.447	0.134
Meteor	0.059	0.536	0.260
GPT Generated (count)	0	2	71

Figure 1.2 - Stage 2 Full Results

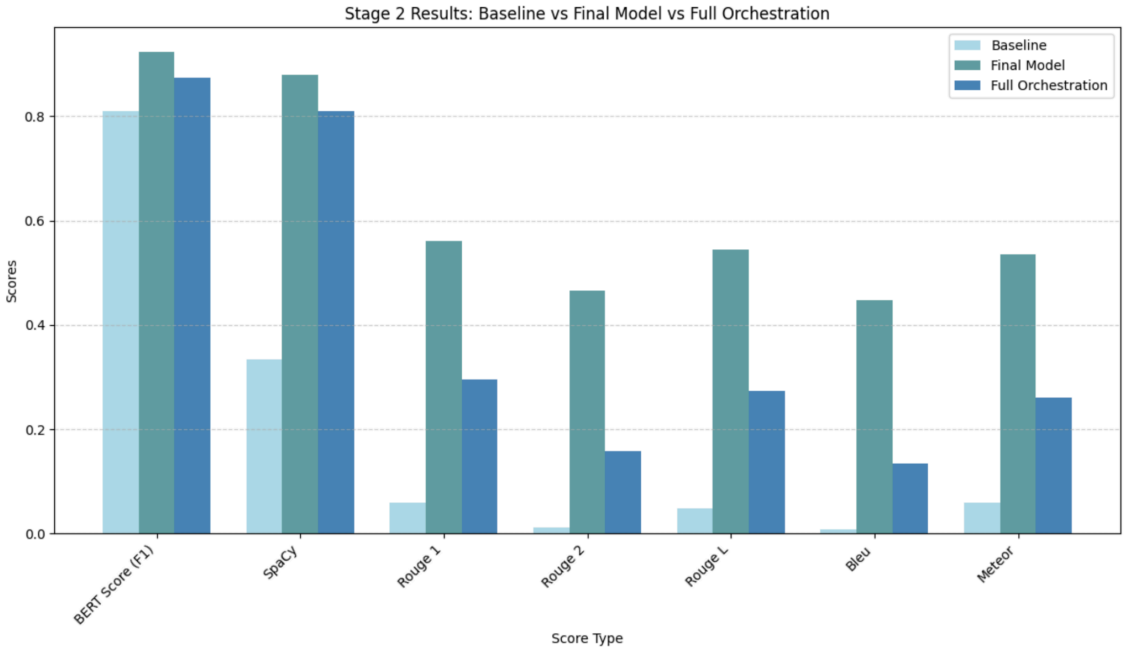


Table 1.3 - Stage 2 Good examples

Highlight provided by Stage 1	Paraphrase	Generated Paraphrase	BERT Score	SpaCy Score
Such information is intrinsically fluctuant and may be inaccurate, incomplete or outdated. Waze does not provide	The service does not guarantee accuracy or reliability of the information	The service does not guarantee accuracy or reliability of the information provided	1	1

any warranties to such informations credibility or reliability. Locationbased Service.	provided			
Legal age depends on the national legislation applicable to the User probably you need to be 18 years old.	This service is only available to users over a certain age	This service is only available to users over 18 years of age	.961	.895
Allscripts reserves the right to restrict who is eligible for an account. Allscripts reserves the right in its sole discretion to determine who may qualify for an account and reserves the right to reject or revoke any account at any time without liability. Allscripts may enable you to create accounts for minors or other members of your family over whom you have legal authority.	Your account can be deleted without prior notice and without a reason	The service can delete your account without prior notice and without a reason	.958	.930

Section 2 - Bias

The results below depict some biases in the dataset and model performance due to the varying distribution of segment types and the differences in their respective scores. The Privacy Policy and Terms segments dominate the dataset, while segments like Cookie Policy, Other, Data, and FAQ are notably underrepresented in the training.

Table 2.1 - Stage 2 Bias

Document Type	Training Count	Testing Count	Stage 1 > 2 Output Count	BERTScore Mean	BERTScore Median	SpaCy Mean	SpaCy Median
Privacy Policy	2,304 (48%)	565 (48%)	464 (48%)	0.87	0.87	0.80	0.81
Terms	1,956 (41%)	493 (42%)	408 (42%)	0.87	0.85	0.82	0.83
Cookie Policy	232 (5%)	57 (5%)	48 (5%)	0.87	0.86	0.78	0.78
Other	186 (3%)	48 (4%)	38 (4%)	0.88	0.87	0.81	0.82
Data	36 (<1%)	9 (<1%)	8 (<1%)	0.88	0.88	0.74	0.73
FAQ	2 (<1%)	1 (<1%)	1 (<1%)	0.87	0.87	0.85	0.85

When examining the scores, the BERTScore Mean is consistently high across all segments, ranging between 0.87 and 0.88. The SpaCy Mean and SpaCy Median, however, have more variability. As expected, the Privacy Policy and Terms segments have high mean scores of .80 and .82, respectively, demonstrating its ability to effectively paraphrase these segments. The lowest results are shown for the Data documentation type. Considering this segment represents less than 1% of the training dataset, this segment could likely be improved by adding more training data for this document type. Despite the large

differences in training data for each of these segments, there appears to be consistency in the performance of the model across all document types.

Figure 2.2 - Distribution of BERTScores by Segment Type

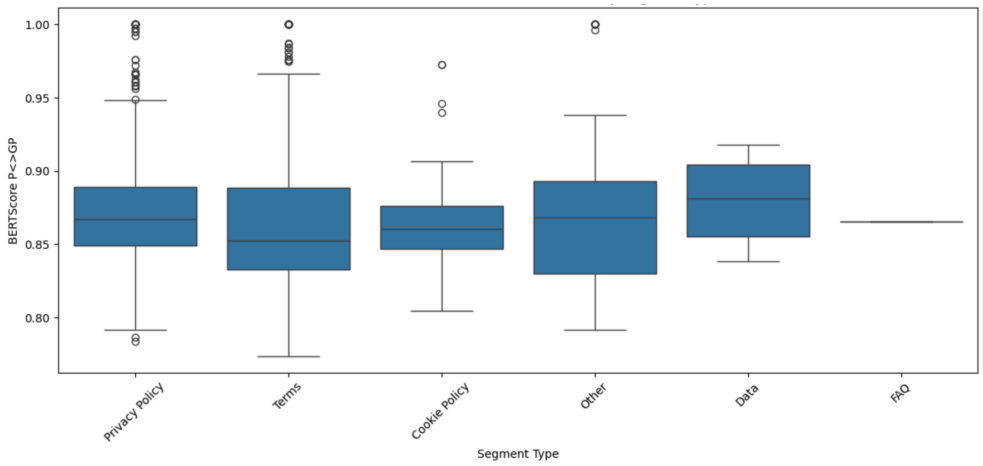
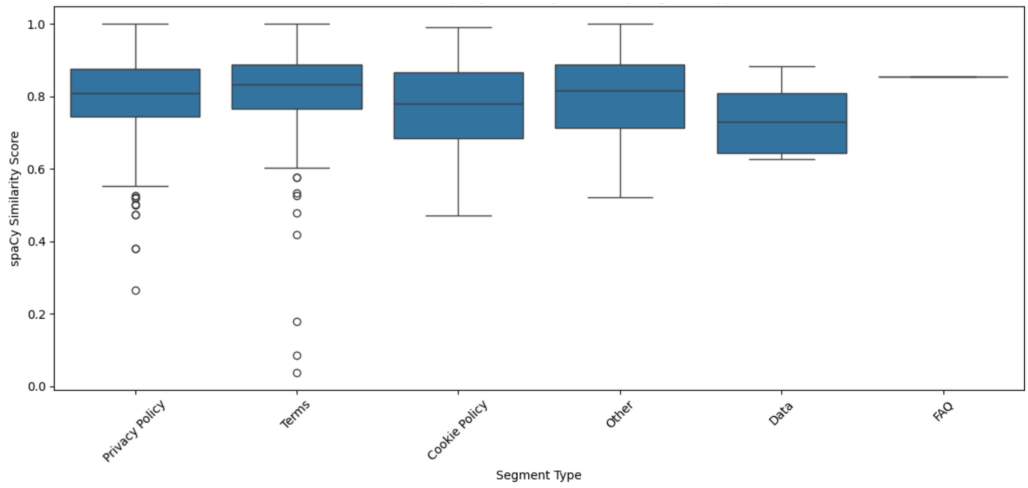


Figure 2.2 - Distribution of SpaCy Scores by Segment Type



Appendix C - Next Steps

Section 1

To enhance this model in the future, there are some key steps that can be taken to improve quality and consistency of outputs. Sun et al. (2020), demonstrate that attention scores capture importance more so than attention weights. To enhance the performance of sentence contribution outlined in this paper, additional experimentation utilizing attention scores could be performed. Table 3.1 below outlines some of the most common paraphrasing errors.

Table 3.1 - Stage 2 Common Errors

Error Type	**Paraphrase**	**Generated Paraphrase**	**BERT Score**	**SpaCy Score**
Over Complication	The service has a no refund policy	This service assumes no liability for any losses or damages resulting from any matter relating to the service	.85	.75
Wrong Audience	This service requires first-party cookies	The service provides information about how they intend to use your personal data	.88	.76
Wrong language	Users agree not to use the service for illegal purposes	El usuario debe proporcionar información veraz al registrarse	.78	.09
Incorrect	Spidering or crawling is not allowed	The service may use your personal data for marketing purposes	.77	.66

Improved data cleaning could help reduce the noise in the input data. There were a few instances of languages other than English in both the training and test sets. While the training data consistently returned paraphrases in English, the both the T5 and the GPT-4o-mini models still returned unexpected results. The T5 model occasionally guessed translations seemingly at random, and the GPT-4o-mini model often returned paraphrases in the same language as the input. In future iterations, other languages would be removed, hyperfocusing on returning correct results in English.

Additional training would help to address some of the other common errors. One frequent issue was the model over-complicating paraphrases by adding details that are not included in the highlight. Continuing the training would help the model learn to only return the most essential information. More training could improve the model's ability to recognize the correct audience for the paraphrase. In some cases the model was returning a summary such as "The service provides information about how they intend to use your personal data" instead of specifying the specific requirement of the service.

Lastly, this model could be improved by lowering the threshold for falling back to the fine-tuned GPT-4o-mini model. Currently, less than 10% of the results were generated using the GPT model in order to minimize costs. This model proved very successful in its results despite being fed only the hardest to handle highlights. Allowing the GPT model to be used more frequently could further enhance overall paraphrase quality without significantly compromising cost efficiency.

By focusing on data cleaning, additional training, and refining the waterfall threshold, future iterations of the model can deliver more accurate, concise, and context-appropriate paraphrases.