Our foray into the hackathon initiative commenced with a meticulous examination of the dataset, focusing on a comprehensive understanding and thorough analysis. The initial step involved the purification of the "train-set.json" file, wherein we employed regular expressions (regex) to eliminate extraneous whitespaces and unwanted characters.Through entity and relations we found out somethings like relations between person and organization with plaintiff and agreement. Subsequently, attention was devoted to rectifying inaccuracies stemming from accent conversions within the dataset.

To delve deeper into the dataset and uncover potential relationships between various entities, we employed advanced techniques, including Conditional Random Fields (CRF) and Parts of Speech (POS) tagging. These methods provided valuable insights into the structure of the data, aiding in the identification of potential augmentation opportunities.

The preliminary exploratory phase paved the way for the application of summarization techniques. We began with established approaches such as TextRank and Latent Semantic Analysis (LSA), leveraging GloVe embeddings within TextRank. However, the results proved to be less than satisfactory, yielding neither useful nor insightful outcomes. Although we could find out the highest frequency words, most important sentences and the sentences with highest relations to the important words.

Subsequently, we turned to the implementation of spaCy, a powerful natural language processing library, for generating summaries. Employing an extractive methodology, spaCy exhibited commendable performance on an open-source dataset, aligning well with the existing original texts. However, when applied to the training dataset, discrepancies emerged. The generated summaries mirrored the original text verbatim, rendering the methodology inefficient for our specific objectives.

Upon careful consideration, we opted to evaluate the efficacy of the summarization process using Rouge scores. Utilizing an evaluation script with the spaCy-generated output for the open-source dataset and the open-source dataset itself as the reference, we observed significantly poor Rouge scores. This discrepancy arose from the fact that the reference summaries did not incorporate words from the original_text corpus, whereas the generated summaries utilized these words and prioritized highly weighted sentences. Consequently, we made the strategic decision to forego the use of the open-source dataset and the spaCy method for the automatic summarization procedure, redirecting our efforts towards alternative methodologies.

After an intensive research of the current state of the art methodologies, we narrowed down to generate summaries using large language models by Facebook and Google. More specifically - facebook/bart-large-cnn, facebook/bart-large-xsum, google/pegasus-large and google/pegasus-big-patent. As these models were pre-trained on legal policy datasets and are most relevant to the expert annotated data, which stood to be the key point of reference in our tasks. The results on provided evaluation script, are shown below:

	Learning rate	Rouge-score-1			Rouge-score-2			Rouge-score-L		
		recall	precision	f1	recall	precision	f1	recall	precision	f1
facebook/bart_large_xsum (pre finetuning):		0.4562091924	0.3740057359	0.3806273182	0.2096241524	0.1746579683	0.1756426063	0.3587381518	0.3015320839	0.302753597
facebook/bart_large_cnn		0.4383654285	0.8389169964	0.5443063579	0.306525244	0.6116134141	0.3846669074	0.3828913482	0.7485269588	0.4788952297
google/pegasus-large(3 epochs)	5.00E-05	0.6330347191	0.5725063163	0.5747643935	0.4311028779	0.3976743922	0.3898381659	0.5991010351	0.5417482487	0.5438633789
google/pegasus-big_patent	5.00E-05	0.5795679307	0.4852398462	0.5128876197	0.3683814647	0.2870746914	0.3116073669	0.5414175261	0.4523304686	0.4786760982
sshleifer/distilbart-xsum-6-6		0.3067921543	0.2336207397	0.2458167815	0.05322722246	0.03790870365	0.04077490264	0.2283928447	0.17593733	0.1833788739
facebook/bart-large-xsum(after finetuning)	5.00E-05	0.9304154635	0.8764046198	0.8907376273	0.883235048	0.8311719871	0.8458544202	0.9131296311	0.8623322832	0.8759557908
google/pegasus-large(10 epochs)	5.00E-05	0.901578229	0.9159459541	0.9001427667	0.853071019	0.8591021428	0.8497796079	0.8840251668	0.8973837197	0.8828568319
google/pegasus-large(10 epochs)	1.00E-04	0.9248798905	0.9180467471	0.9147117181	0.8778127628	0.8699739363	0.8691066617	0.9073970116	0.901797018	0.8984859245

Next step: Experimentation on model training. A number of parameters were fine tuned to optimize the scoring metrics, which involved varying different-

- Learning rates
- Batch size and epochs
- Choice of optimizer
- Hidden layers count
- Max_length and length penalty.

The training was conducted on Euro Lex sum dataset (https://huggingface.co/datasets/dennlinger/eur-lex-sum/tree/main), mainly because of its similarity with expert data in terms of summary length and extractive approach of summarization. The results, although were not satisfactory and marked significantly less than the models when trained and tested on airbus-helicopters-train-set.

As a result, we moved forward with finalizing the “google/pegasus-large” model, with the most optimized training parameters for best scores.