CS230

Information Extraction from Parole Hearing Transcripts

Courtney McGill* cmcgill@stanford.edu
Dan Xie* danxie26@stanford.edu
Kaikai Sheng* kaikais@stanford.edu

Abstract

We explored two NLP paradigms, i.e. question answering and Question-answering NLI (QNLI), to extract different information from long parole hearing transcript. We built a DistilBERT-based QA model to tackle a case factor which is factual-based. It achieved achieve an F1 score of 96.12 and an EM score of 95.39. We further explored QNLI paradigm by fintuning it with different pretrained models on datasets annotated by us. We found out that the combination of positive annotations and negative annotations play the most important role in model performance. The model achieves a hit rate of 84% and 66.7% for validation set on the two case factors we examined. This is a more than 20% improvement over the keyword-based heuristic approach, and the performance can extend to other case factors.

1 Introduction

California holds thousands of parole hearings for eligible prisoners each year. In each hearing, an up-to 150-page transcript of the entire conversation is produced for the government and public to review. Studying the parole hearing transcripts is a key part of understanding the criminal justice apparatus and of reviewing individual cases, but the length and quantity of these hearings means it would require immense human resources to examine and compare these cases. Because of this, these transcripts and the accompanying parole decision are rarely audited.

Our team explored a variety of NLP methods to extract context that provide answers for case factors that are important to parole decision-making from the aforementioned transcripts. We chose this project because it is both socially impactful and technically challenging. Its technical challenges lie in following three folds.

- **High Text Length**: The parole hearing transcripts have around 15000 words on average. Such long text document poses a significant challenge for state-of-the-art NLP model like BERT [1] which can only take at most 512 tokens as input.
- **Very Few Suitable Training Data**: While there are many public datasets available for NLP research, few of them are dialogue-based, especially for information-extraction task. On top of that, we only have 567 annotated transcripts out of all 35105 transcripts and annotations are not well suited for information-extraction task.
- **Diversity and Vaguity**: Case factors varies a lot from one to another, some of which are factual-based and always appear in specific positions of transcript, e.g. Minimum Eligible Parole Date (MEPD), while some of which are more open-ended and vague, e.g. attorney opinion.

*Equal contribution. Mentored by Jenny Hong (jyunhong@stanford.edu) and Drew A. Hudson (dorarad@cs.stanford.edu)

CS230: Deep Learning, Winter 2021, Stanford University, CA.

Given above challenges, we decided to tackle the problem in two NLP paradigms, i.e. framing it in two different ways.

- **Question Answering (QA)**: We first built a DistilBERT-based (a distilled version of BERT) [2] QA model to tackle case factor, MEPD, which is factual-based and always appears in the opening statement of a parole hearing transcript. Our model achieved **F1** score of 96.12 and **EM** score of 95.39.
- **Question-answering NLI (QNLI)**: We also built a DistilBERT-based QNLI model to tackle case factors that are more open-ended and vague and can appear anywhere in the transcript. The model will find the context span from transcript that provides answer to the case factor. Its top-pick hit rate is more than 20% higher than heuristic keyword search method.

2 Dataset

Through a prolonged legal process with the California Department of Corrections and Rehabilitation (CDCR), we have obtained a comprehensive set of all parole hearing transcripts from 2007-2019, 35,105 in total. A subset of these hearing transcripts were used to make the primary dataset for *Project Recon*, thus requiring it to remain a private.

The dataset is composed of probation hearings and corresponding labels that capture important information from the hearing. Each hearing is specific to an individual that is serving a life-sentence in California but was eligible to be released on parole at the time of their hearing. Each hearing is made up of metadata such as the names of people participating in the trial, the prison, etc. as well as the hearing proceedings and the parole decision. The hearing itself is dialogue primarily between the commissioner and parole candidate but also includes occasional statements from the attorney and any other participants.

Although 567 of these cases do have coded labels, the extractive nature of our project required us to do additional labeling to denote the exact answer spans for MEPD and context sentences for count 155s and risk assessment. Due to the limited amount of training data, we also used a few other public dataset for transfer learning, which we will talk about in later sections. We summarize the key stats of datasets in Table 6.

3 Approach

3.1 QA task

For our first task we mainly explored the DistilBERT model [2] for a QA task. We chose selected this model because it is a fast, lightweight model that has historically performance on various NLP tasks. To enable DistilBERT-based QA model to perform on parole hearing QA tasks, as shown in Figure 5, we first fine-tuned it in a combinations of three large public QA datasets, i.e. SQuAD [3], NewsQA [4], Natural Questions [5], each of which has 50000 training data points (we will call it task fine-tuning stage for the rest of context since its primary purpose is to adapt DistilBERT to QA task).

We then continue fine-tune the model in the small parole hearing annotated datasets (we will call it domain fine-tuning stage for the rest of context since its primary purpose is to adapt task fine-tuned model to perform domain-specific QA tasks), which has 567 training data points. The goal is to better adapt model to perform parole hearing dataset, which has very different data distribution than aforementioned three public datasets. The case factor in parole hearing dataset does not come with questions. For case factor of interest, i.e. minimum eligible parole date, we come up with *anchor questions* that is intended to get answers to case factor.

Given the excellent performance of DistilBERT for our task, which will be discussed in the experiments section, and the little performance increase historically offered by using BERT (appendix D.), we decided not to explore any other architectures for the QA task.

While encouraged by this excellent result, we believe that this result is not generalizable across all case factors given that we are aided by knowing to look specifically in the opening statement of the transcript as well as by many of opening statements having a sentence that explicitly calls out inmate's MEPD in one dialogue turn, e.g. "Inmate received a term of – (cough) excuse me – of15 to

life plus two with a minimum eligible parole date of June 28th, 2000.". To tackle case factors that may appear anywhere in the transcript and are not consistently identified explicitly, we shifted our framing of this problem from a QA paradigm to QNLI one.

3.2 QNLI Task

For this second task, we built a model that searches for context span (about 300-token in length) that contain answers to case factors. The entire pipeline diagram is shows in 5. We focused on the case factors of **count 115s (disciplinary marks)** and **psychiatrist risk assessment** since they are two of most important case factors for parole decisions.

- We task-finetuned different pretrianed models on QNLI dataset [6] (derived from SQuAD [3]) to see how it performs. We select ones to continue to explore on Recon dataset by making trade-off between finetuning/eval speed and model accuracy. The purpose of task-finetuning is to get the model to adapt to QNLI task since we hava a very limited amount of Recon training data for task-finetuning.

  We explored GPT-2 [7] as an example of a more traditional language model, which consists of only the decoder part of a traditional transformer and is unidirectional.

  The superior results of DeBERTa were unsurprising given that its distinctive features, a disentangled attention mechanisim and an enhanced mask decoder, have lead to improved performance over BERT and RoBERTa on several NLP tasks [8].

  Given the strong results of the DeBERTa model, as well as the fast training time of DistilBERT, we decided to move forward with these two models for tuning on our private parole hearing transcript.

- We annotated 567 transcripts to create finetuning dataset and eval dataset by: 1) adding context spans that provides answers to case factors as positive examples 2) adding context spans that cannot provide answers to case factors, esp. tricky ones with keywords that are related to case factors, as negative examples.

- We then domain-finetuned the task-finetuned model on aforementioned annotated dataset. We tuned the hyperparameters of domain-finetuned model only with **count-115** validation dataset and test it on held-out **count-115** test dataset. We then applied the same model and model setting on **risk assessment** dataset to see how the model performs on case factor and data that are its out-of-domain to test its generalization power.

4 QA Experiments

4.1 Experiment Setup

Throughout the entire section, we use a default hyper-parameter setting of $num\_epochs = 3$, $batch\_size = 16$, $learning_rate = 3e - 5$ and train on a AWS VM equipped with NVIDIA Tesla V100 GPU unless otherwise specified. We obtained the scaffolding code from RobustQA codebase [2]. We split our annotated parole hearing dataset (567 data points in total) into train dataset (367 data points), dev dataset (100 data points) and test dataset (100 data points). We use the two standard metrics of Exact Match (EM) and F1 Score (F1) to measure QA model performance.

4.2 Effect of Domain Fine-tuning, *Anchor Questions* and Data Cleaning

As we see from Table 1, noting that #anchor questions = 0 means no domain fine-tuning, domain fine-tuning significantly improve the model performance. However, number of anchor questions does not matter as long as there are anchor questions for domain fine-tuning. Data cleaning does not either since the QA model is robust enough to handle uncommon wording pattern like name spelling (e.g. R-O-B-E-R-T-S) or prompt (e.g. [sic]).

[2] https://github.com/MurtyShikhar/robustqa

# Anchor Questions	With Data Cleaning		Without Data Cleaning	
	F1	EM	F1	EM
0	57.36	54.72	58.23	55.73
2	96.24	95.62	96.93	96.52
4	97.08	96.63	96.79	96.52
6	96.23	95.84	96.83	96.29
8	95.56	94.49	96.31	95.96
10	96.12	95.39	96.22	95.73

Table 1: The relationship between **F1/EM**, number of anchor questions and data cleaning. Note that #anchor questions = 0 means no domain fine-tuning.

4.3 Experiment Results

As we see from Table 1, our QA model achieves great results for MEPD case factor on opening statement. The model achieves **F1** score of around 96 and **EM** score of around 95.

We believe this excellent result is because many of opening statements have a sentence that explicitly call out inmate's MEPD in one dialogue turn, e.g. "Inmate received a term of – (cough) excuse me – of15 to life plus two with a minimum eligible parole date of June 28th, 2000.". We analyzed loss patterns in appendix B.

5 QNLI Experiments

5.1 Experiment Setup

The experiment setup is similar to QA experiments since both experiments use BERT variant pretrained models except

- We split our annotated parole hearing dataset (567 data points in total) into train dataset (417 data points), dev dataset (75 data points) and test dataset (75 data points) since we need more training data for finetuning.
- Inspired by ImageNet contest top-5 accuracy metrics, we use top-K hit rate as measurement metrics which measures accuracy of model's top-K picks, i.e. a model prediction is considered to be correct is any of its top K picks contains the annotated context span.

We chose this metric since it better aligns with the end goal of applying NLP to do information extraction from transcripts. Although it cannot pinpoint the exact answer from transcript like extractive QA model (often there is extractive answer span for open-ended and vague question), it does add value for downstream processing, e.g. applying T5 on extracted context span to do abstractive question answering.

5.2 Experiment Results

Based on prior experiment results of QA model, we did not experiment with data cleaning techniques and anchor question choice. We mainly focus on experiment with model choice and data we use for finetuning. We experiment with different settings only on count_115s since we want to find a setting that achieves better performance than baseline and its performance is able to extend to most of other case factors if not all case factors. The risk_assessment acts as a held-out test benchmark for this purpose.

We chose heuristic keyword search as the baseline since heuristic search is what Recon project team uses currently (though it is not exact the same heuristic search as Recon team's). The heuristic keyword search approach find context span by weighted-sum of number related keywords in the context span.

As it is shown in the Table 2,

- Finetuning and the data for finetuning plays a significant role for model performance. The model performs well only if finetuned with right combination of negative examples and postivie examples. Otherwise, it performs worse than heuristic approach.
- DistilBERT model performs best if finetuned with both positive examples and negative examples that combined with tricky ones and randomly-chosen ones. The top-1 hit rate is more than 20% higher than heuristic approach.
- The good performance extends to risk_assessment case factor, which is more vague and open-ended than count_115s. The top-1 hit rate of DistilBERT is 20% higher than heuristic approach. To our surprise, the top-3 hit rate is also almost 20% higher than heuristic one. We speculated that is is because 1) heuristic approach is more susceptible to choice of keywords. 2) The risk assessment context is spread across a long dialogue which is difficult for heuristic approach to work well but it is not the case for count_115s, which usually the parole officer will call out like "your total count of 115s is xxx".
- DeBERTa shows no statiscally better performance than DistilBERT. We will not explore it for further research.

Case Factor	Model	Finetuning Data	top-1	top-2	top-3	top-4	top-5	top-6
count_115s	Heuristic	NA	63.7	85.3	96.0	97.3	97.4	97.4
count_115s	DistilBERT	none	57.3	74.7	81.3	82.7	85.3	88.0
count_115s	DistilBERT	pos only	8.0	20.0	20.7	36.0	40.0	48.0
count_115s	DistilBERT	pos+tricky neg	57.3	77.3	82.7	85.3	90.7	92.0
count_115s	DistilBERT	pos+2*random neg/case	72.0	90.7	98.7	98.7	98.7	100.0
count_115s	DistilBERT	pos+5*random neg/case	73.3	92.0	97.3	97.3	98.7	98.7
count_115s	DistilBERT	pos+tricky neg+ 2*random neg/case	84.0	93.3	94.7	97.3	98.7	98.7
count_115s	DeBERTa	pos+tricky neg+ 2*random neg/case	85.3	96.0	98.7	100.0	100.0	100.0
risk_assess	Heuristic	NA	42.7	58.7	73.3	84.0	89.3	94.7
risk_assess	DistilBERT	pos+tricky neg+ 2*random neg/case	66.7	85.3	94.7	94.7	96.0	97.3

Table 2: Comparison of model different settings for two case factors by top-K hit rate on **validation** dataset. Heuristic method is to search for context span by weighted keywords. The "pos" means positive annotation, i.e. ones annotated with "entailment". The "tricky neg" means tricky negative annotation, i.e. ones that has keywords but annotated with "not entailment". The "2*random neg" means negative examples annotated with "not entailment" that are randomly chosen from transcript.

Case Factor	Model	Finetuning Data	top-1	top-2	top-3	top-4	top-5	top-6
count_115s	Heuristic	NA	64.0	82.7	89.3	96.0	96.0	97.3
count_115s	DistilBERT	pos+tricky neg+ 2*random neg/case	88.0	97.3	100.0	100.0	100.0	100.0
risk_assess	Heuristic	NA	44.0	68.0	80.0	89.3	92.0	94.7
risk_assess	DistilBERT	pos+tricky neg+ 2*random neg/case	60.0	77.3	90.7	92.0	93.3	93.3

Table 3: Comparison of model different settings for two case factors by top-K hit rate on **test** dataset. Heuristic method is to search for context span by weighted keywords. The "pos" means positive annotation, i.e. ones annotated with "entailment". The "tricky neg" means tricky negative annotation, i.e. ones that has keywords but annotated with "not entailment". The "2*random neg" means negative examples annotated with "not entailment" that are randomly chosen from transcript.

6 Conclusion

We explored two NLP paradigms, i.e. question answering and Question-answering NLI (QNLI), to extract different information from long parole hearing transcript. The QA model achieved an F1 score of 96.12 and an EM score of 95.39 but it is only applicable to case factor which is factual-based. The QNLIs models achieves more than 20% hit rate than keyword-based heuristic approach, and the performance can extend to other case factors. The extracted context span by QNLI can be further explored downstream tasks, e.g. abstractive question answering.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[2] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019.

[3] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics.

[4] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[5] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, March 2019.

[6] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2019.

[7] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2018.

[8] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention, 2021.

[9] Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. A discrete hard em approach for weakly supervised question answering, 2019.

[10] Sathish Reddy Indurthi, Seunghak Yu, Seohyun Back, and Heriberto Cuayáhuitl. Cut to the chase: A context zoom-in network for reading comprehension. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 570–575, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[11] Raghavendra Pappagari, Piotr Żelasko, Jesús Villalba, Yishay Carmiel, and Najim Dehak. Hierarchical transformers for long document classification, 2019.

[12] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California, June 2016. Association for Computational Linguistics.

A Contributions

Courtney primarily work on model task-finetuning, data processing and annotation. Dan primarily work on model visualization, data processing and annotation. Kaikai works on model experiments, data processing and annotation.

B Related Work

Information extraction over long formed text is challenging research area in NLP. Usually, the task is break down into two steps: the regional proposal to extract relevant spans from the long formed text, and then perform NLP task over the relevant spans. For the regional proposal step, with very limited train data available, some research chose the first or random relevant span, while others explored improvement with different methods, which includes token based similarity method, EM approach [9] and Reinforcement Learning approach [10].

Besides the two-step approach, other researches model the end to end process. ToBERT [11] breaks long formed documents into chunks and models the sequence of chunk representations from BERT with a small Transformer. Hierarchical Attention Networks [12] take sequence of words as input, then utilizes sequence model with attention mechanism to maintain and aggregate relevant local information from word level to sentence level.

C Data Preprocessing

C.1 QA Task

Given our initial goal of extracting the MEPD from the opening statement, our primary data task was to extract and format the opening statement. Since the opening statement is often delivered by the commissioner at the beginning of the hearing, we used the heuristic rule described below to extract the opening statement:

1. We considered the intersection of the first 10 statements and the first 5 statements delivered by the commissioner to be the statements that were eligible for selection as the opening statement.

2. Statement with the maximum number of characters was selected as the opening statement.

3. However, if the minimum eligible parole date is not mentioned in the selected statement, we instead take the concatenation of the candidate statements.

Additionally we cleaned the opening statement by removing non-alpha numeric and punctuation characters from the statement.

C.2 Context Detection

Given our second goal of context sentence detection, the first pre-processing step was reformatting the transcript dataset to appear simply as a list of sentences with accompanying unique identifiers. We manually annotated minimum context spans for 574 transcripts, which were used as fine-tuning set in addition to QNLI data set. Sentences, which contain direct answer to the target case factor, are consider positives.

D DistilBERT vs. BERT Comparison

This collection of tables is copied from the original DistilBERT paper [2].

Table 1: **DistilBERT retains 97% of BERT performance.** Comparison on the dev sets of the GLUE benchmark. ELMo results as reported by the authors. BERT and DistilBERT results are the medians of 5 runs with different seeds.

Model	**Score**	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI
ELMo	68.7	44.1	68.6	76.6	71.1	86.2	53.4	91.5	70.4	56.3
BERT-base	79.5	56.3	86.7	88.6	91.8	89.6	69.3	92.7	89.0	53.5
DistilBERT	77.0	51.3	82.2	87.5	89.2	88.5	59.9	91.3	86.9	56.3

Table 2: **DistilBERT yields to comparable performance on downstream tasks.** Comparison on downstream tasks: IMDb (test accuracy) and SQuAD 1.1 (EM/F1 on dev set). D: with a second step of distillation during fine-tuning.

Model	IMDb (acc.)	SQuAD (EM/F1)
BERT-base	93.46	81.2/88.5
DistilBERT	92.82	77.7/85.8
DistilBERT (D)	-	79.1/86.9

Table 3: **DistilBERT is significantly smaller while being constantly faster.** Inference time of a full pass of GLUE task STS-B (sentiment analysis) on CPU with a batch size of 1.

Model	# param. (Millions)	Inf. time (seconds)
ELMo	180	895
BERT-base	110	668
DistilBERT	66	410

E Pretained Models

Model Architecture	**Pretrained Implementation**
GPT-2	distilgpt2
DistilBERT	distilbert-base-uncased
RoBERTa	roberta-base
DeBERTa	microsoft/deberta-base

Table 4: Huggingface models used and tested for use.

F Model performance after task-finetuned on QNLI dataset

Model	**Accuracy**
GPT-2	84.50
DistilBERT	89.33
RoBERTa	90.41
DeBERTa	**92.13**

Table 5: Accuracy of various model architectures when applied to public QNLI dataset.

G Analysis on Loss Patterns

We spot two common loss patterns when the QA model get MEPD answer wrong.

In the following loss example, the QA model hallucinate an answer probably because the predicated answer starts with a number.

- **Question**: What is date that you will be eligible for parole?

- **Context**: Forty. The victim in this case was [redacted]. The hearing is being recorded. For the purpose s of voice identification, each of us will state our first and last name, spelli ng our last name. When it's your turn , Mr. [redacted], after you spell your last name you're going to give your CDC R number. I'm going to begin. We're going to go around the room to my left . We'll end up with you. My name is [redacted], Commissioner with the Board of Parole Hearings..
- **Answer**: NA
- **Prediction**: Forty. The victim in this case was [redacted]

In the second loss example, the QA model probably got confused when two dates are close to keywords "possible release date".

- **Question**: What is date that you will be eligible for parole?
- **Context**: All right. We're on the record, and this is in the matter of [redacted]. Today is an Initial Suitability Hearing. Today's date is March 3, 2015. ... So he was totally sentenced for 23 years. He has an earliest possible release date of August 27, 2020, and a youthful offender release date of November 12, 2012. The victim in the robbery case was [redacted]. Now the hearing is being recorded, so for the purposes of voice identification, we're going to go around the room and identify ourselves ...
- **Answer**: August 27, 2020
- **Prediction**: November 12, 2012

H Example of Opening Statement of parole hearing transcript

"Okay. We're on the record. This is a subsequent hearing number three for inmate [redacted]. CDC Number [redacted]. Today's date is [redacted], 2007. This hearing is being held at [redacted] State Prison. The time is now 0900 hours. The inmate was received on [redacted] 1988, from Los Angeles County. The life term began [redacted], 1988, with a minimum eligible parole date of June 28th, 2000. The controlling offense is Second Degree Murder with Use of a Firearm. The case number is [redacted]. As to Count One, PC [redacted] Use of a Firearm Enhancement. Inmate received a term of – (cough) excuse me – of 15 to life plus two with a minimum eligible parole date of June 28th, 2000 – yes, the year 2000. This hearing is being recorded. For the purpose of voice identification, each of us will state our first and last name, spelling our last name for the record. When it is your turn, [redacted], please after you've spelled your name, state your CDC Number for the record. I will start and go to my right. [redacted], Commissioner of Board of Parole Hearings.",

I Data statistics of parole hearing annotated transcripts

Number of Transcripts	567
Avg. Number Words per Transcript	15,399
Avg. Unique Speakers per Transcript	12.68
Avg. Speaking Turns per Transcript	711.66

Table 6: Data statistics of parole hearing annotated transcripts.

Labeled Recon	Average Length	StdDev Length	Max Length	Contains MEPD
Max Of Early Commissioner Statements	183.60	116.68	1827.00	484/563
All Early Commissioner statements	277.73	174.78	2834	510/563

Table 7: Data statistics of opening statement of parole hearing annotated transcripts.

J Data statistics of opening statement of parole hearing annotated transcripts

K Attention Analysis

We conducted analysis below to understand how the model attends to relevant content in different layers and attention heads. In this example, we use sample question "What is the risk assessment for the inmate?", and a random sentence in training set "However, moving to page 10, under the overall risk assessment, the doctor writes that after weighing all the data from the available records, the clinical interview, the risk assessment data, it's opined that you present a relatively moderate risk for violence in the free community." as input. In this case, the correct answer can be derived from the sentence.

Figure 1. demonstrates how the question as whole attends to individual sub-word in the sentence, in 9th layer of the fine-tuned DeBERTa model. We observed that the head 6 attends the most to the correct answer span ("a relatively moderate risk for violence") in the sentence. Zooming into the head 6 in Figure 2, the specific token that drives overall attention to the correct answer span is "?". Although "?" itself does not carry any semantic meaning of the question, however considering it's at the 9th layer towards the end, it's very likely that "?" has aggregated the semantic meaning of the question to some degree.

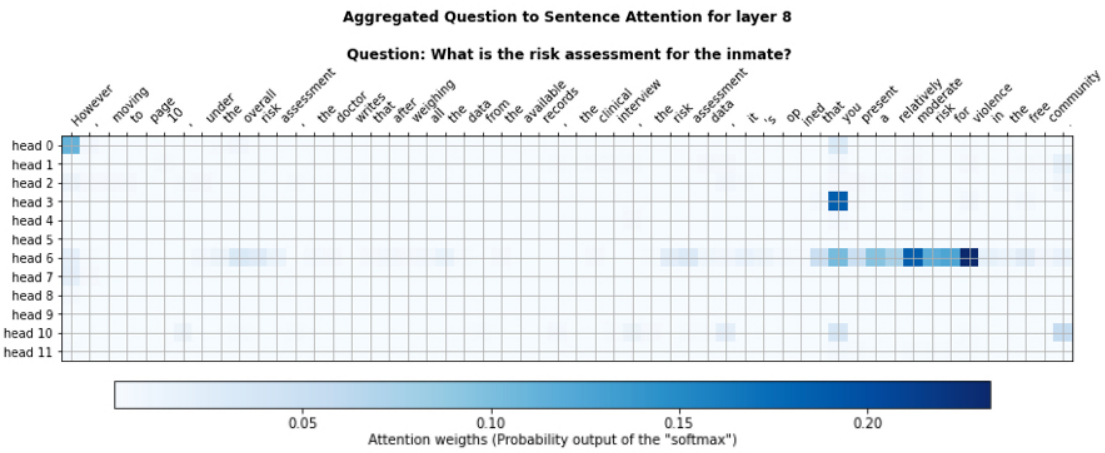

Figure 1: Aggregated Question to Sentence View

The reversed view in Figure 3 demonstrates how the sentence as whole attends to the question, in order to understand what the question is asking. Head 1 stands out and its attention is focused on the "risk", "assessment", "inmate" and "the", which covers the key words that convey the meaning of the question.

Figure 4 displays a similar question to sentence attention view, but for in layer 0. It's clear that head 0 attends majorly to the key words "risk" and "assessment", which are highly relevant to the semantic

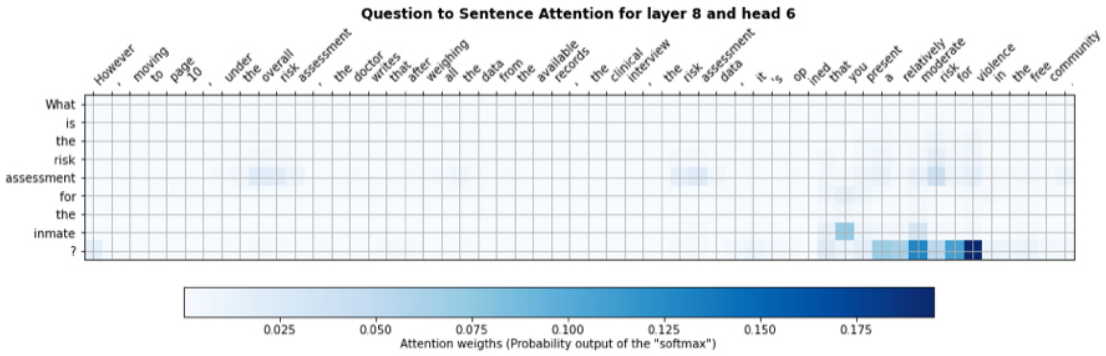

Figure 2: Token Wise Question to Sentence View

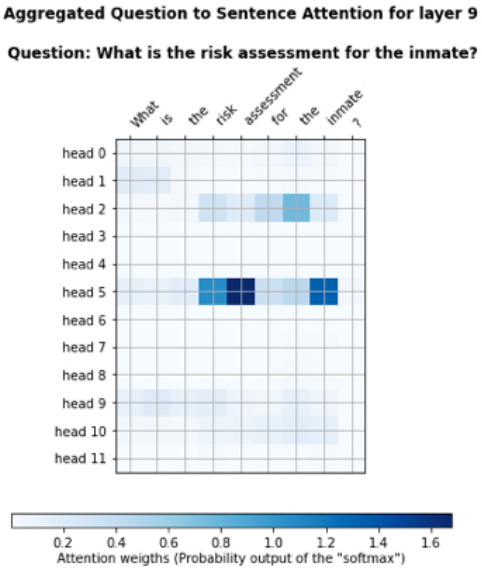

Figure 3: Aggregated Sentence to Question View

meaning of the question; head 2 attends greatly to punctuation, while head 4 focuses on stop words; head 8 and head 11 behaves as bag of word attention.

The comparison between attention pattern in layer 0 and the layer 8 , confirms the idea that it's similar to how Computer Vision deep neural network works, the earlier layers in the deep network gathers more granular features while the latter layers learn more aggregated and sophisticated features based on granular ones.

L Model Training Pipeline Diagram

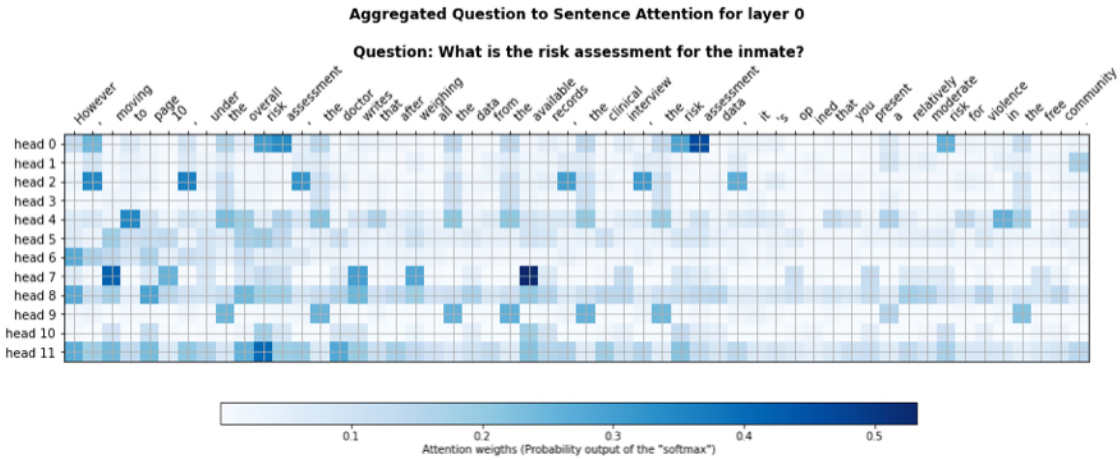

Figure 4: Aggregated Question to Sentence View

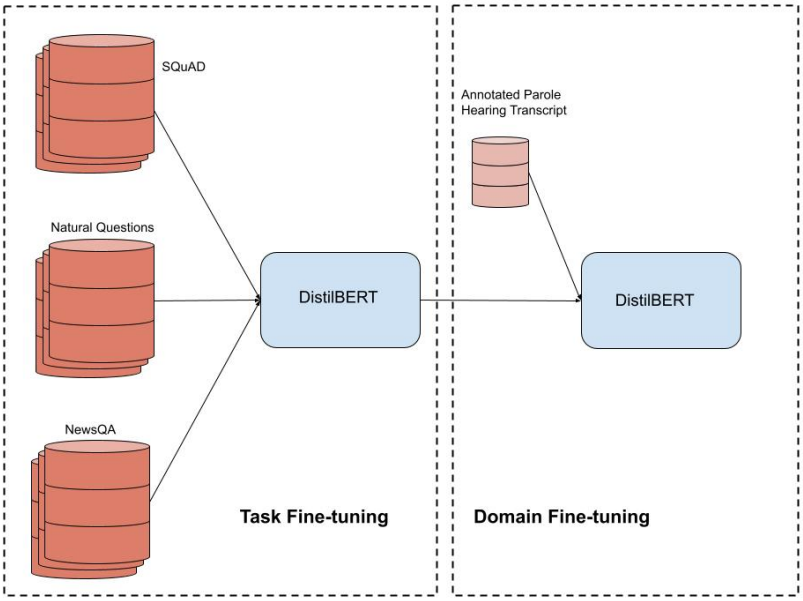

Figure 5: The QA model training diagram for the parole hearing dataset. We task fine-tuned the DistilBERT-based QA model on three large public QA datasets. We then domain fine-tuned on annotated parole hearing dataset.

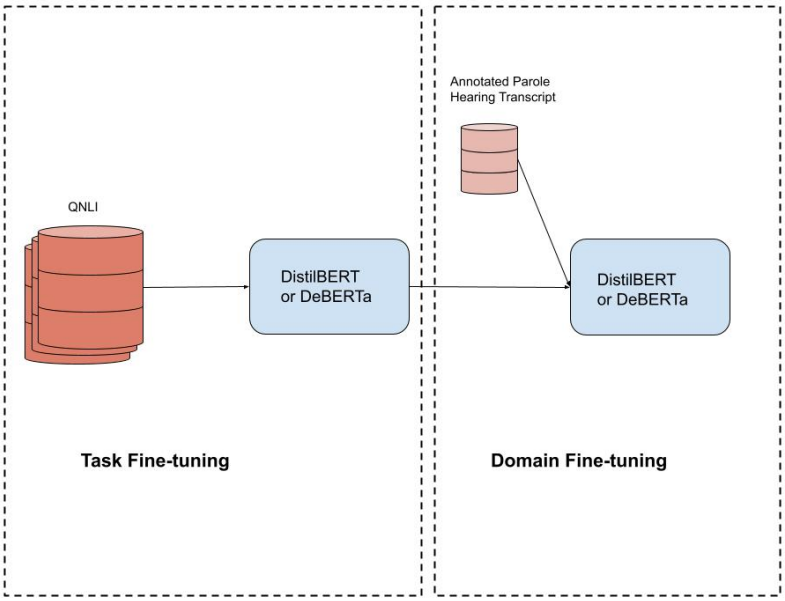

Figure 6: The QA model training diagram for the parole hearing dataset. We task fine-tuned the DistilBERT-based QA model on three large public QA datasets. We then domain fine-tuned on annotated parole hearing dataset.