Towards Evaluating the Complexity of Sexual Assault Cases with Machine Learning

CSE 350/450: AI for Social Good

Bruke Mammo, Praveer Narwelkar, Roshan Giyanani
CSE Department, Lehigh University
{bruke.mammo, pnarwelkar, roshangiyanani} @gmail.com

I. Introduction

AEquitas, as part of the Sexual Assault Justice Initiative (SAJI) with the Justice Management Institute, and Urban Institute want to do performance analysis on sexual assault cases. As part of this process, humans are asked to review cases and pull out information such as the age of the victim, the victim's gender, ethnicity, etc. Already stretched thin, justice departments do not have the spare 2.5 hours/case to accomplish this task. (AEquitas, the Justice Management Institute, & the Urban Institute, 2018)

We propose using Natural Language Processing (NLP) tools such as SDNet, created by Microsoft, and datasets like SQuAD, from Stanford, to aid in extracting these features for justice departments and SAJI (Rajpurkar, Jia, & Liang, 2018; Zhu, Zeng, & Huang, 2018).

This paper serves as the final report for Lehigh University's AI for Social Good class (CSE 350/450).

II. Terminology

Accuracy Measures

- **F1 score =** 2 * [(precision * recall) / (precision + recall)]
    - Performance metric commonly used in machine learning, from 0 to 100.
- **Precision = #** true positive results / (# true positive results + # false positive results)
    - Performance metric from 0 to 100, the percentage of relevant results.
- **Recall =** # true positive results / (# true positive results + # false negative results)
    - Performance metric from 0 to 100, the percentage of relevant results classified correctly.

- **Accuracy** = (# true positive results + # true negative results) / # data entries
    - Performance metric, the percentage of results classified correctly.

Natural Language Processing Overview

Our work makes use of multiple different models which attempt to solve two of the hottest problems in the domain of natural language processing, which are language modelling and question-answering. Before we actually get into the models themselves, it is essential to describe these problems in some depth as they are a crucial part of any text analytics project.

Language Modelling

Language modelling is the first step in any NLP application. It is the process of designing the probabilistic distribution of the language present in the text. Using this probabilistic distribution, we can easily predict the occurrence of a particular word in a sentence (or a sentence itself) after a set of words/sentences have already occurred. Some terms that are related to language modelling and would be found throughout the report:

- <u>Context:</u> This is the text which contains the sentence/word that is expected to be the answer.
- <u>Token:</u> This is the term to define any uniquely occurring word in a sentence or simply define the non-repeating phrases/sentences. It is the smallest unit of the language model.
- <u>Vocabulary:</u> This is the set of all words present as well as absent from the given context. This is the dictionary that is referred to by the model when it needs to compute the probabilities.

Question-Answering

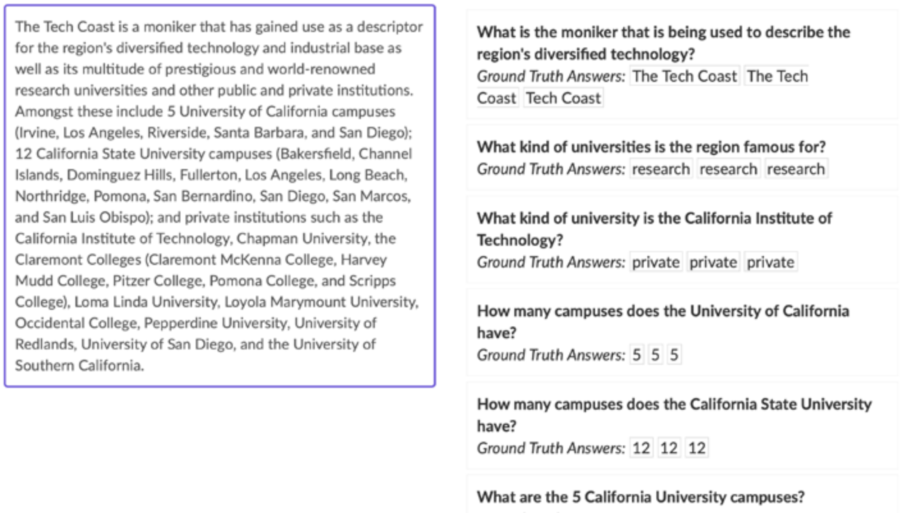
Question-Answering, or QA, is a growing field of research, as it is highly required for chatbots, as well as any system that is designed to automatically answer questions posed by humans using sentences generated in a natural language format. The types of questions targeted by this research is a wide range, including the following types:

- Factual (Did)
- List (How many)
- Definition (What)
- Reasoning (Why)
- Hypothetical (What if)
- Semantically constrained (certain conditions imposed upon the core question)
- Cross-lingual (multiple of the above types present in the same question)

These QA systems (often one of the underlying implementations of a chatbot aimed at answering customer enquiries) utilize a structured knowledge base of possible questions and their correct and plausible answers. Using this knowledge base as the reference, they can pull answers from the accompanying unstructured contexts accompanying the questions (a collection of natural language documents). Internally, the QA system will form a language model based on this knowledge base and the related contexts to be able to generate valid sentences with the proper content as the answers. We will be discussing a few of the most accurate QA systems currently published in academic research. As the complexity of the questions increases, so does the difficulty in obtaining a valid answer. This is clearly visible through the design process of these systems.

III. Related Work

SQuAD: Stanford Question Answering Dataset



The Tech Coast is a moniker that has gained use as a descriptor for the region's diversified technology and industrial base as well as its multitude of prestigious and world-renowned research universities and other public and private institutions. Amongst these include 5 University of California campuses (Irvine, Los Angeles, Riverside, Santa Barbara, and San Diego); 12 California State University campuses (Bakersfield, Channel Islands, Dominguez Hills, Fullerton, Los Angeles, Long Beach, Northridge, Pomona, San Bernardino, San Diego, San Marcos, and San Luis Obispo); and private institutions such as the California Institute of Technology, Chapman University, the Claremont Colleges (Claremont McKenna College, Harvey Mudd College, Pitzer College, Pomona College, and Scripps College), Loma Linda University, Loyola Marymount University, Occidental College, Pepperdine University, University of Redlands, University of San Diego, and the University of Southern California.

What is the moniker that is being used to describe the region's diversified technology?
Ground Truth Answers: The Tech Coast | The Tech Coast | Tech Coast

What kind of universities is the region famous for?
Ground Truth Answers: research | research | research

What kind of university is the California Institute of Technology?
Ground Truth Answers: private | private | private

How many campuses does the University of California have?
Ground Truth Answers: 5 | 5 | 5

How many campuses does the California State University have?
Ground Truth Answers: 12 | 12 | 12

What are the 5 California University campuses?

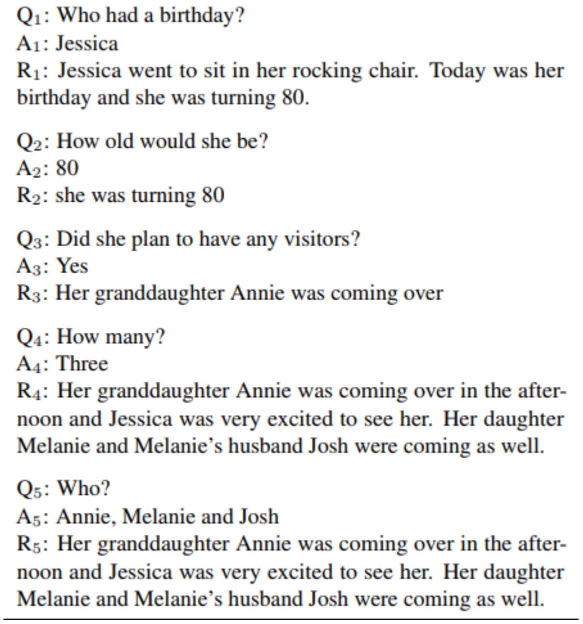
Figure 1: A screenshot excerpt from the SQuAD website. It describes how the process works.

SQuAD contains short texts from wikipedia, called “contexts”, and a dataset of 100,000 questions, answered by humans. These answers are in the form of a quote from the text. SQuAD is used to train and evaluate an NLP model’s ability to answer reading comprehension questions. (Rajpurkar, Jia, & Liang, 2018)

Rajpurkar, Jia, & Liang later created SQuAD 2.0, which added 50,000 questions with no possible answers. This can be used to train models to abstain from answering unanswerable questions. Humans get an F1 accuracy of 89.45% when answering SQuAD 2.0, and the best models have a comparable accuracy of 89.47%.

CoQA

______CoQA stands for Conversational Question Answering Challenge (Siva Reddy, Danqi Chen, Christopher D. Manning, August 2018). The CoQA dataset is designed to enable machines to answer conversational questions - a series of interconnected questions and answers. The answers provided in this set are free-form text with evidence marked out from the actual context. By adding the conversational format of QA, it introduces two important and challenging aspects to the machine, coreference and pragmatic reasoning. SQuAD-trained models are unable to answer questions requiring better comprehension abilities, such as questions based on complex semantic constraints and certain categories of factual questions starting with “did”. This difference in question distribution is discussed later in Methodology - SDNet and CoQA . CoQA makes use of free-form text answers and a “rationale”, which usually is the text evidence from the related context (Figure 2).



Q_1: Who had a birthday?
A_1: Jessica
R_1: Jessica went to sit in her rocking chair. Today was her birthday and she was turning 80.

Q_2: How old would she be?
A_2: 80
R_2: she was turning 80

Q_3: Did she plan to have any visitors?
A_3: Yes
R_3: Her granddaughter Annie was coming over

Q_4: How many?
A_4: Three
R_4: Her granddaughter Annie was coming over in the afternoon and Jessica was very excited to see her. Her daughter Melanie and Melanie’s husband Josh were coming as well.

Q_5: Who?
A_5: Annie, Melanie and Josh
R_5: Her granddaughter Annie was coming over in the afternoon and Jessica was very excited to see her. Her daughter Melanie and Melanie’s husband Josh were coming as well.

Figure 2: A screenshot excerpt from the CoQA website. It describes how the process works, plus we add a description of how it is different from SQuAD.

The conversation proceeds as a sequence, which begins with the context, followed by the question with its answer (which could be implicit), followed by the “rationale”, or the actual (or closest) text span from which the answer is found by the model and rephrased (not simply extracted unlike many other QA systems). The answering process is more natural and not as extractive as other QA systems, resulting in answers that contain the meaning of the selected text span and the context of the question In addition, CoQA learns from the previous questions in the conversation sequence.

This further allows the model to provide more natural answers, particularly for single worded questions which wouldn’t otherwise have any answers in the text. Another great point about CoQA is its ability to find multiple valid answers for a single question in the given context/rationale. This allows the dialog agents to have more of an impact on the conversation by hitting more correct answers. We have utilized this aspect while trying to answer the question about the relationship between the victim and the suspect.

CoQA has roughly 127k questions grouped into 8k conversations (each conversation based on a different passage has about 5 questions). These passages are selected from a wide range of domains (5 in-domain evaluations and 2 out-of-domain evaluations).

BERT

BERT stands for Bidirectional Encoder Representations from Transformers. It is a novel language modelling technique published by Devlin, Chang, Lee, & Toutanova (Google) in October 2018. Traditional language modelling techniques go from left to right or right to left (start to end, which forms the forward pass probability distribution over a sentence; or end to start, which forms the backward pass probability distribution). And the distribution becomes better with higher size of the token used in the modelling process (speaking in terms of the n-gram modelling). What makes BERT different and so much better from the traditional methods is the way it trains a language model. It implements an attention mechanism called the Transformer to learn the contextual relationship between words or sentences in a given context. This transformer contains an encoder and a decoder; only encoder is needed for this task. The encoder scans the entire sequence of words (token or sentences) in one go, therefore making the language model bidirectional in nature (more precisely, non-directional). Every word is compared with the entirety of the surrounding context (and not just a few words in one direction), to form a knowledge graph of how the word is related with all the entire context.

BERT creates its language model by defining the prediction target using two unsupervised training strategies, “Masked Language Modelling” (MLM) and “Next Sentence Prediction” (NSP). In MLM, upto 15% of the tokens in every sequence (or sentence) are randomly selected and then replaced using a MASK token. Then the sequences are fed into BERT. The model is then tasked with predicting the original, unmasked values of these tokens based on the remaining unmasked tokens in every sequence. This essentially gives a probability distribution of the possible values for the masked tokens. One tradeoff here is that BERT converges slowly as its loss function ignores non-masked tokens; but this improves the context awareness, therefore resulting in better predictions. On the other hand, in NSP, the model is trained to be able to distinguish between two sentences. The task that this training enables is the computation of the chance that the second sentence in the input pair is the subsequent sentence of the first one in the input. The training process is split into two parts - first being the 50% of the inputs to the model being a pair of consecutive sentences, and the second being the remaining 50% of the inputs containing a pair of random sentences from the context. The inputs are preprocessed by adding token, sentence and positional embeddings in the sentences and then passed on to the model. “CLS” and “ESP” tokens are used to mark the start and end of a sentence, respectively. The pre-trained BERT Base model is created by using both of these strategies simultaneously, and the combined loss is the objective function that is to be minimized. Any application of BERT is a simple layer add-on over this base model.

SDNet

SDNet is an innovative contextualized attention-based deep neural network published by C Zhu, M Zeng, and X Huang from the Microsoft Speech and Dialogue Research Group, in December 2018. This model is introduced with the intent to improve the comprehension of conversation over a context and extract relevant information from the provided passage. It makes use of inter-attention and self-attention techniques together on the passage and questions to achieve a good and more effective understanding of the context itself as well as the conversation history. This model is inspired by machine reading comprehension and conversational question answering tasks. This model makes use of BERT and the history of questions asked (along with their answers). This model has three main parts -

- Encoding layer: It forms a fixed length vector by encoding each token from the context and the questions. This vector includes the word embeddings and the contextualized embeddings. The contextualized embeddings are obtained from BERT's output, which is achieved by using a fixing the parameters of the BERT model and then computing a weighted linear sum of the embeddings from different layers in BERT.
- Integration Layer: It employs a batch of multi-layer recurrent neural networks (RNNs) for capturing the contextual information from the passage and the questions, as RNN is the most suitable for a sequence data. The output hidden vectors are generated after passing through a dimensionality reduction procedure. Self-attention techniques are utilized for drawing out the relationships between the words at different positions in the context and the question.
- Output Layer: It computes the final answer span. The attention calculated in the integration layer is used to condense the question into a fixed-length vector, which then provides the probability of the start and end positions of the answer after subjection to a bilinear projection.

Now that we have covered the models and related background knowledge in sufficient depth, it is time to formulate the problem statement.

IV. Problem Statement

As part of SAJI, AEquitas wants to gather data on sexual assault case outcomes using a a new form. These forms contain details such as the name of the victim, the location of where the incident took place, the age of the victim, etc. The information will come from incident or case reports. Our goal is to demonstrate we can answer a subset of these questions, focusing on four target categories in particular, due to their varying complexity, variety of question types, and presence in our data.

Victim Age

The age of the victim is the easiest question to answer, where the goal is to extract a piece of text containing the age. For example, text containing the phrase "I was 18" should return "18" when we ask the model to determine the age. If no age is mentioned, then the true answer is considered to "unknown."

Victim Consumption of Drugs and Alcohol

The consumption of drugs and alcohol are more difficult categories. It's not as simple as extracting a matching piece of text. Here, we want a binary yes or no answer. For example, "I had been drinking vodka” has an expected answer of “yes” for if the victim consumed alcohol, and similarly for drugs. If no mention of consuming alcohol or drugs is present in the text, then the true answer is considered to be “unknown.”

Victim and Suspect Relationship

The relationship between the victim and the suspect is the most difficult question to predict the answer for due to the complexity of it, as we move from a binary prediction to a multi-class prediction. Potential results are as follows: stranger, family member, brief encounter, non-stranger, current or former intimate partner, or professional. If none of these are defined in the text, then the true answer is considered to be “unspecified” or “unknown.”

V. Methodology & Experiments

Data

Source

Due to the sensitive nature of the case reports, it is impossible to access them outside of a controlled setting, which means we do not have direct access to a real dataset for training and testing purposes. To work around this, we had to find a pseudo dataset that we could use to demonstrate the capabilities of our proposed solution. The ideal pseudo dataset would be both similar in content type and content subject, which means we looked for data that was qualitative text about sexual assault incidents. While we found plenty of quantitative, statistical data about such incidents, and found plenty of qualitative text corpuses dealing with other sorts of events (e.g. news datasets), we did not successfully find a dataset that combined the two features. Initially, we began working with a BBC News corpus that dealt with five categories of stories: sports, technology, politics, entertainment, and business (Greene & Cunningham, 2006).

After a conversation with Professor Sihong Xie from Lehigh University, a Professor of Computer Science who deals with natural language processing, we were warned away from using this data. Although it is qualitative text, it doesn’t deal with the subject matter we want and would cause problems for us later in the project. For example, with news stories about business, we would never be able to answer the question of whether or not a victim had been drinking alcohol. He pointed us towards using data from Reddit, a forum website where nearly any manner of conversation can be found.

After confirming with the Reddit Terms of Service and their Privacy Policy, we followed the suggestion and built our own dataset of 410 comments pulled from Reddit’s API through PRAW, the Python Reddit API Wrapper. These comments were from threads where sexual assault survivors shared and discussed their stories. The Terms of Service and Privacy Policy state that comments fall under the category of User Content, which are acceptable for our use as long as we do not modify, publish, profit off or encourage illegal activity with it, none of which we are doing.

To protect anonymity and for privacy purposes, we deleted the local copy of data created from the Reddit data after the completion of the course.

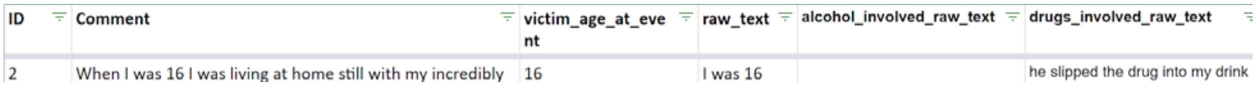
Labelling

After downloading the data, we built a CSV file where each comment was an entry on each row. In the following columns, we manually marked up the data for the categories that we wanted to answer. We currently do not label any implicitly mentioned answers for the selected categories, in the contexts. The CSV format can be seen below, in Table 1.

Title	**Data Type**	**Description**
id	Integer	identification number for the comment
comment	String	full text of the comment
victim_age_at_event	String	true answer for the age of the victim
raw_text	String	context text for the victim_age_at_event
alcohol_involved_event	String	true answer for if the victim consumed alcohol
alcohol_involved_raw_text	String	context text for the alcohol_involved_event
drugs_involved_event	String	true answer for if the victim consumed drugs
drugs_involved_raw_text	String	context text for the drugs_involved_event
relationship_victim_suspect	String	the relationship between the victim and the suspect
relationship_victim_suspect _raw_text	String	context text for relationship_victim_suspect

Table 1: Dataset variables and their descriptions.

If there were multiple places in the comment where an answer could be selected, we included them all with a pipe character (|) to denote separate answers. For example, a potential combination could be "yes|yes" for alcohol_involved_event and "I was home to drink | got me drunk" for alcohol_involved_raw_text.



ID	Comment	victim_age_at_event	raw_text	alcohol_involved_raw_text	drugs_involved_raw_text
2	When I was 16 I was living at home still with my incredibly	16	I was 16		he slipped the drug into my drink

Figure 3: A screenshot excerpt from the dataset to demonstrate what it looks like. The full comment will not be shown due to the sensitive nature of the data.

Frequency/Distribution

The distribution of the reddit data we manually labelled can be seen in Figure 4. Starting from the top left chart, it shows the involvement of drugs on the victim's side in sexual assault; to be more precise, whether or not a victim had consumed drugs prior to the sex offense committed against her/him. It can be a willing or unwilling consumption. Only a small percentage of the total comments mention drugs. For the sake of simplicity we consider forced or willing consumption of drugs as a single category. From the data we observed there were certain cases where the victim wasn't sure they were drugged. We considered this a "not drugged" case (No answer); again for the sake of simplicity.

Then coming to the top right chart, we have the demographics for the consumption of alcohol by the victim before the assault. Again, for the sake of simplicity we are not breaking down alcohol consumption into multiple varieties of alcohols. We are simply labelling the data as whether the victim had drunk alcohol or not, willingly or unwillingly. Most of the contexts didn't have mention about alcohol in them.

The bottom left chart talks about the age of the victim at the time of assault. More precisely, whether the age of the victim at the time of the assault was mentioned in the context or not. Roughly a little less than half of the contexts mentioned the victim's age at the event. We considered age mentioned in words as a positive mention of age. We didn't, however, consider age ranges or implicit mention of ages (such as early 20's or "three years later" or the second summer after X years old etc). Finding a way to understand implicit mentions of age and time is important, but out of scope for our purposes.

Lastly, the bottom right chart mentions about the relationships between the victim and the suspect. Here we labelled the data using the categories provided in the SAJI document. We currently have 7 categories labelled: stranger, brief encounter, family member, non-stranger, former intimate partner, current intimate partner, and professional. 2 contexts fall under multiple categories.

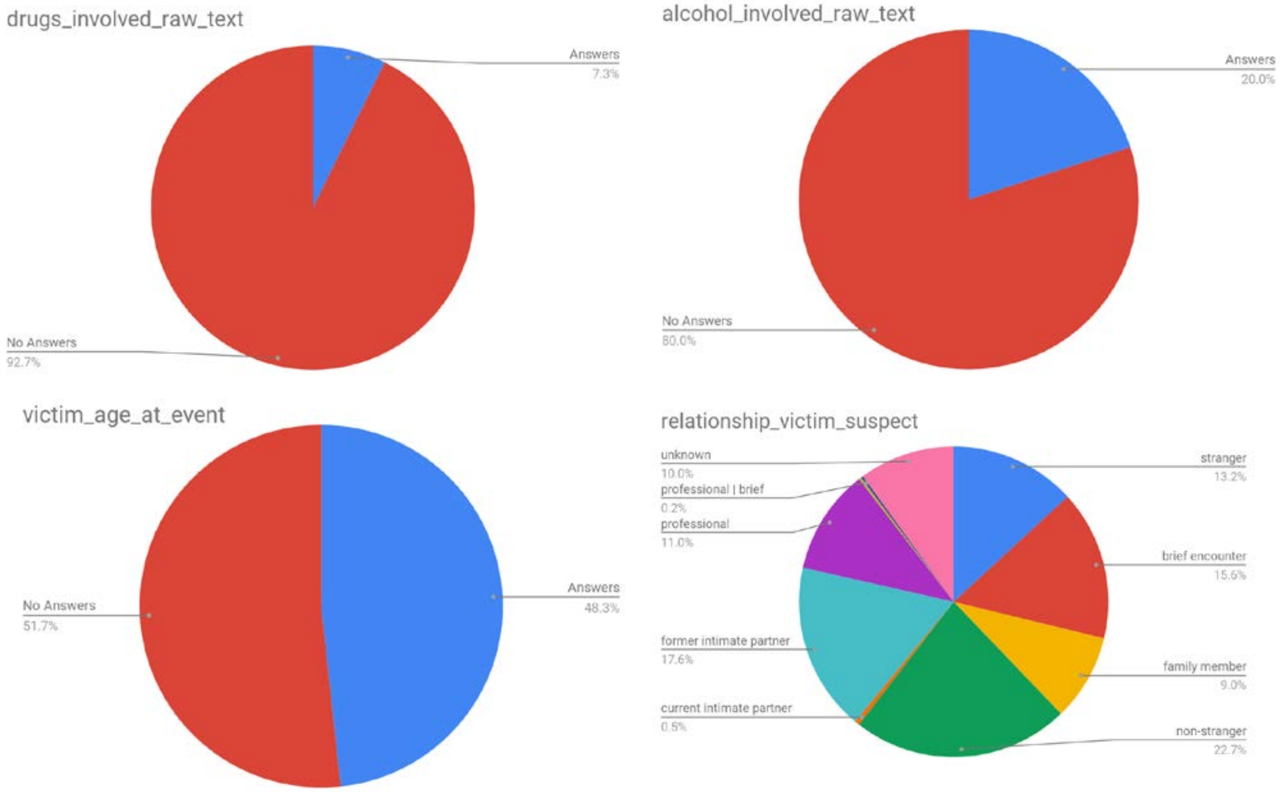


Figure 4: The distribution of labels for, clockwise from top left, whether the victim consumed drugs, whether the victim drank alcohol, how the victim knew the suspect, and the age of the victim at the time of the event.

BERT + SQuAD

Our initial plan was to ask BERT, fine tuned on the SQuAD 2.0 dataset, to answer the question “How old is the victim?”. For this to work, though, we first had to finetune BERT on SQuAD 2.0. Since the BERT model is incredibly large and computationally intensive (the README recommends fine tuning the smaller version of BERT on a graphics card with 12gb of VRAM), we had to set up various computing resources and experiment with various hyperparameters.

We eventually settled on using the pretrained BERT-Base, Uncased: 12-layer, 768-hidden, 12-heads, 110M parameters, fine tuned with a train_batch_size of 6, a max_seq_length of 68, a learning rate of 2e-5, and 2 training epochs. The training took approximately 2 days, and our base model reported an f1 of 65.28% on the SQuAD 2.0 dataset. This is not as good as the full version on the leaderboards, but good enough for our purposes. Now that we have trained BERT, we had to ask it our question.

This required us to first convert our labeled data into the SQuAD format BERT could understand. This was a repetitive task we automated with a python script. When we evaluated BERT with the labeled data and the question, “How old is the victim?”, we got an f1 and

HasAns_f1 score of ~8% using about half our data. When we reran on the full data, we got an f1 of 57.71% and 4.24%, respectively. We focus on the second of these scores for age in particular because NoAns_f1 is almost always 100%, or very close to it, and f1 is just a weighted average of NoAns_f1 and HasAns_f1. This was a pretty bad result, but we wondered if we could improve by asking the same question in a different way.

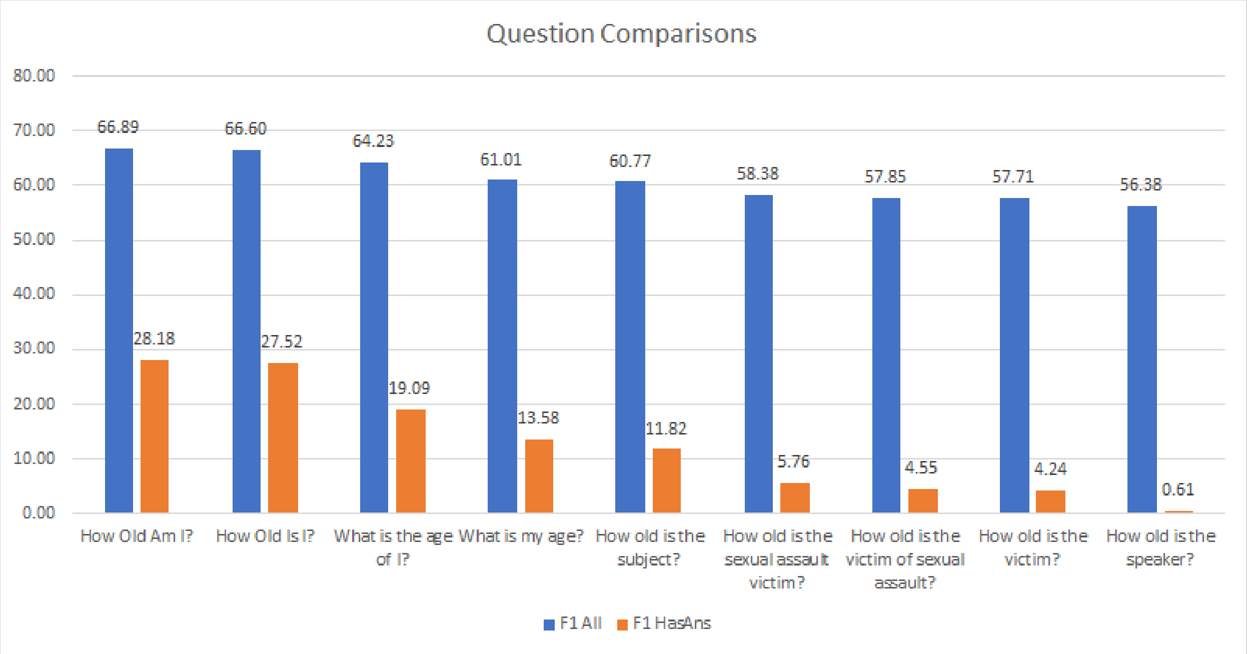


Figure 5: f1 scores for age questions using only a pre-trained Bert model on SQuAD data. "F1 All" refers to the total f1 score for the entire data, "F1 HasAns" refers to the f1 score when considering only data that has a defined true answer..

We sat down, brainstormed a few possible questions, modified our scripts, and evaluated again. There was a big difference, as you can see in Figure 5. We found that specifying victim with the words "sexual assault" improved results a little bit, but since the reddit data is first person, we hypothesized that BERT wasn't able to associate "victim" with the subject. Sure enough, specifying the subject's POV had a large positive impact. Our best result came from the question, "How old am I?", which gave us a HasAns_f1 score of 28.18%.

At this point, we had two questions: (1) Could we improve BERT's predictions by fine-tuning it with some of our data, and (2) How correlated were the answers different questions resulted in?

We first focused on fine-tuning BERT because we thought that would have a bigger impact. Our method for this was, for each question, to split our data into 3 folds, training on 2, and evaluating on the third. We would then alternate which fold we evaluated with, and average the results across the three 3 folds for each question.

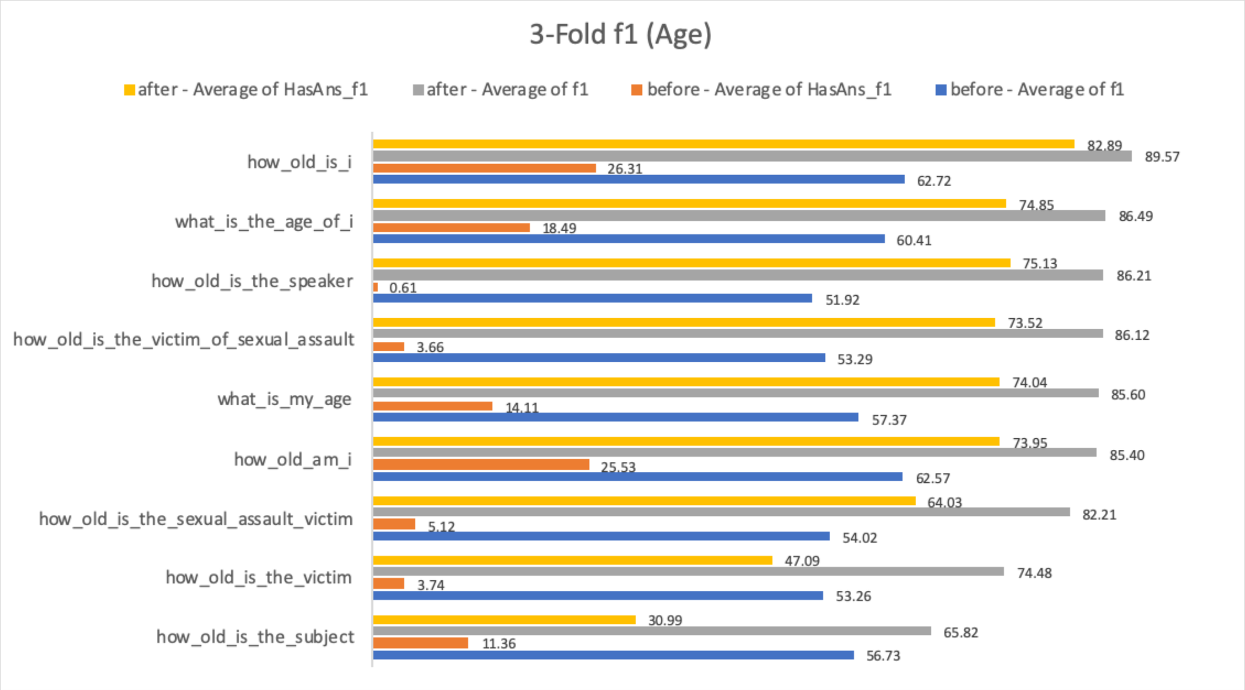
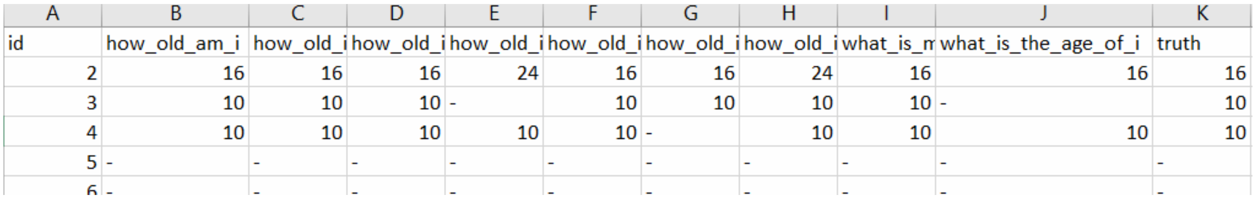


Figure 6: Output for the f1 scores of age questions. "f1" refers to the total f1 score, "HasAns_f1" refers to the f1 score when considering only questions with a defined true answer. "before" refers to using only the pre-trained Bert model with SQuAD data, "after" refers to using the pre-trained Bert mode that has been fine-tuned with our data.

This also improved our HasAns_f1 scores dramatically. Using the same questions, our new worst result, "How old is the subject?", had a higher HasAns_f1 at 30.99% than the best result without training. Our new best result, "How old is I?", had a HasAns_f1 of 82.89%, followed closely behind by 5 other questions, all at a ~74-75% HasAns_f1. Interestingly, we saw that considering POV in our questions had less of a benefit, with "How old is the victim of sexual assault?" in that set of close questions.

Multi-Question Classifier I

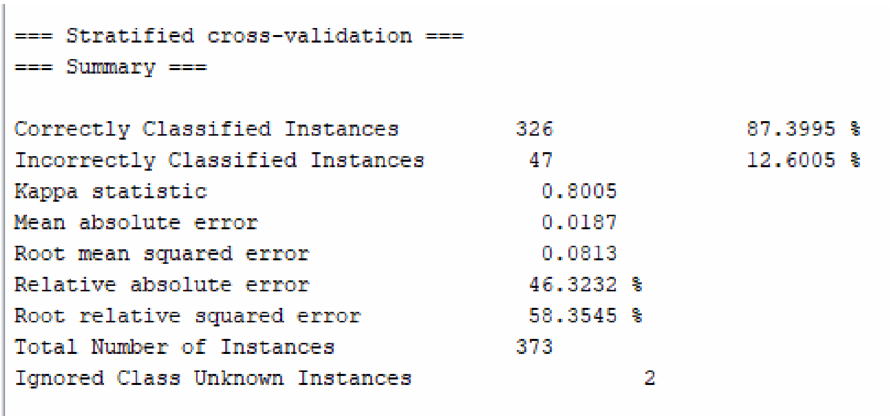
We then returned to our other question about the correlation of predictions across different questions. Once these predictions are made for each category and each question, we can't say for sure that the question with the highest f1 score perfectly subsumes any other question we tested. It's possible that an entry that the highest-performing question predicted incorrectly as "no" for alcohol_involved_event could have been correctly predicted as "yes" by another question. With that in mind, we combine the predictions from each question for a category into a CSV file for that category, each question being a column, the first column being the question's id number and the final column is the true label from our marked up data for that entry. If there are four questions, there would be six rows, and an example row for the "alcohol_involved" file could be "2,yes,yes,unknown,no,yes." This reads as comment #2, the first two questions predicting a "yes," the third unable to find a prediction, the fourth predicting "no," and the correct label being "yes."



	A	B	C	D	E	F	G	H	I	J	K
1	id	how_old_am_i	how_old_i	how_old_i	how_old_i	how_old_i	how_old_i	how_old_i	what_is_m	what_is_the_age_of_i	truth
2	2	16	16	16	24	16	16	24	16	16	16
3	3	10	10	10	-	10	10	10	10	-	10
4	4	10	10	10	10	10	-	10	10	10	10
5	5	-	-	-	-	-	-	-	-	-	-
6	6	-	-	-	-	-	-	-	-	-	-

Figure 7: Example collation file for use in WEKA for the age category. A dash in this images signifies no answer. We have nine possible questions to ask for the age of the victim, for a total of eleven columns.

Afterwards, we use a tool called WEKA, the Waikato Environment for Knowledge Analysis, to turn these individual predictions into a master prediction for each comment. The tool provides a simple-to-use GUI as well as an API to perform data analysis and machine learning tasks. Using the Vote classifier, we aimed to combine the predictions of multiple other nominal-data classifiers and used 4-fold validation to determine whether or not we were able to increase the accuracy over using a singular question on its own. The three WEKA-provided classifiers we used were Logistic, LWL or locally-weighted learning, and RandomTree, with a prediction combination method of Averaging the Predictions.

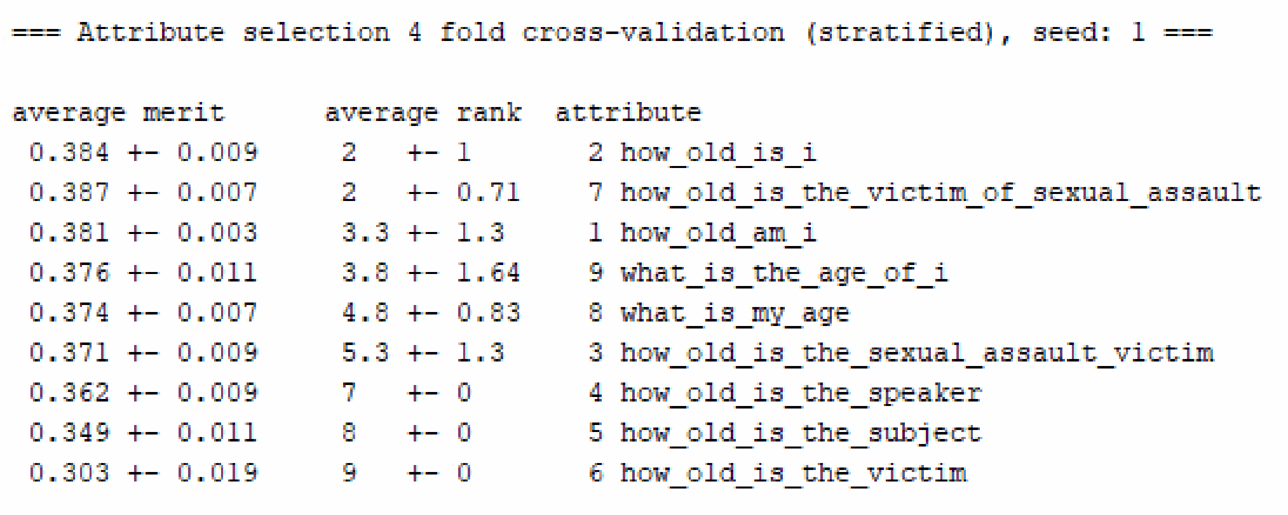


```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         326               87.3995 %
Incorrectly Classified Instances        47               12.6005 %
Kappa statistic                          0.8005
Mean absolute error                      0.0187
Root mean squared error                  0.0813
Relative absolute error                 46.3232 %
Root relative squared error             58.3545 %
Total Number of Instances              373
Ignored Class Unknown Instances                  2
```

Figure 8: Example results of the Vote classifier for WEKA for the Age category. This has an accuracy of 87.40%, so it is not performing better than some questions on their own.

A large problem with this is that WEKA does not have a method for predicting the correct value for the truth out of the values provided by the attribute columns. Ideally, using Figure 8 above as an example, what we want is for the model to identify that “16” is the majority answer and therefore, to predict “16” as the correct value for id “2.” Instead, WEKA treats each value that it sees in true labels as a class, so instead of simply pulling out “16” because there’s a 16 in the attribute columns, it’s trying to learn what combinations will give it a class “16.” While this may seem like the same thing, it means that WEKA would be unable to predict the correct label for an entry where the truth label is an age that it hasn’t seen before. This is largely due to the fact that we treat age labels as strings or nominal data, as both “16” and “sixteen” could be seen in the raw text.

Though WEKA didn’t prove helpful for age, we learned of an additional feature it provides called Attribute Evaluation, where it allows us to determine which one of the attributes has the highest correlation between its value and the value of the true label. In other words, WEKA provides a way to determine which questions, or combinations of questions, are the most effective. This is important as we can eliminate questions that are unnecessary, which speeds up runtime across all aspects (preprocessing, post processing, running). We use the CorrelationAttributeEval evaluator with Ranker for the search method.



```
=== Attribute selection 4 fold cross-validation (stratified), seed: 1 ===

average merit      average rank  attribute
 0.384 +- 0.009     2   +- 1       2 how_old_is_i
 0.387 +- 0.007     2   +- 0.71    7 how_old_is_the_victim_of_sexual_assault
 0.381 +- 0.003     3.3 +- 1.3     1 how_old_am_i
 0.376 +- 0.011     3.8 +- 1.64    9 what_is_the_age_of_i
 0.374 +- 0.007     4.8 +- 0.83    8 what_is_my_age
 0.371 +- 0.009     5.3 +- 1.3     3 how_old_is_the_sexual_assault_victim
 0.362 +- 0.009     7   +- 0       4 how_old_is_the_speaker
 0.349 +- 0.011     8   +- 0       5 how_old_is_the_subject
 0.303 +- 0.019     9   +- 0       6 how_old_is_the_victim
```

Figure 9: Example results of the CorrelationAttributeEval for WEKA for the Age category. A higher "average merit" signals a higher correlation.

In Figure 9 above, we see that the three most important questions that we have for age are "How old is I?", "How old is the victim of sexual assault?" and "How old am I?" This suggests to us that questions that are worded similarly are deemed unnecessary in determining the final value, even if they have a high accuracy on their own. "How old am I?" is one of the lower-performing questions for the age category, with an f1-score of 85.40% and a HasAns_f1-score of 73.95%, but all three of the top attributes here are better than the age baseline performance of 58%.

SDNet + CoQA

Now that we had a pipeline for evaluating the effectiveness of questions, we decided to attempt to extract other pieces of information. We decided this would be asking if the victim ingested alcohol. The baseline accuracy for alcohol is 79%, which we determined by calculating the accuracy for when "no" was predicted for every question. We came up with a long list of questions, and ran our pipeline again. This time, though, we had very bad results.

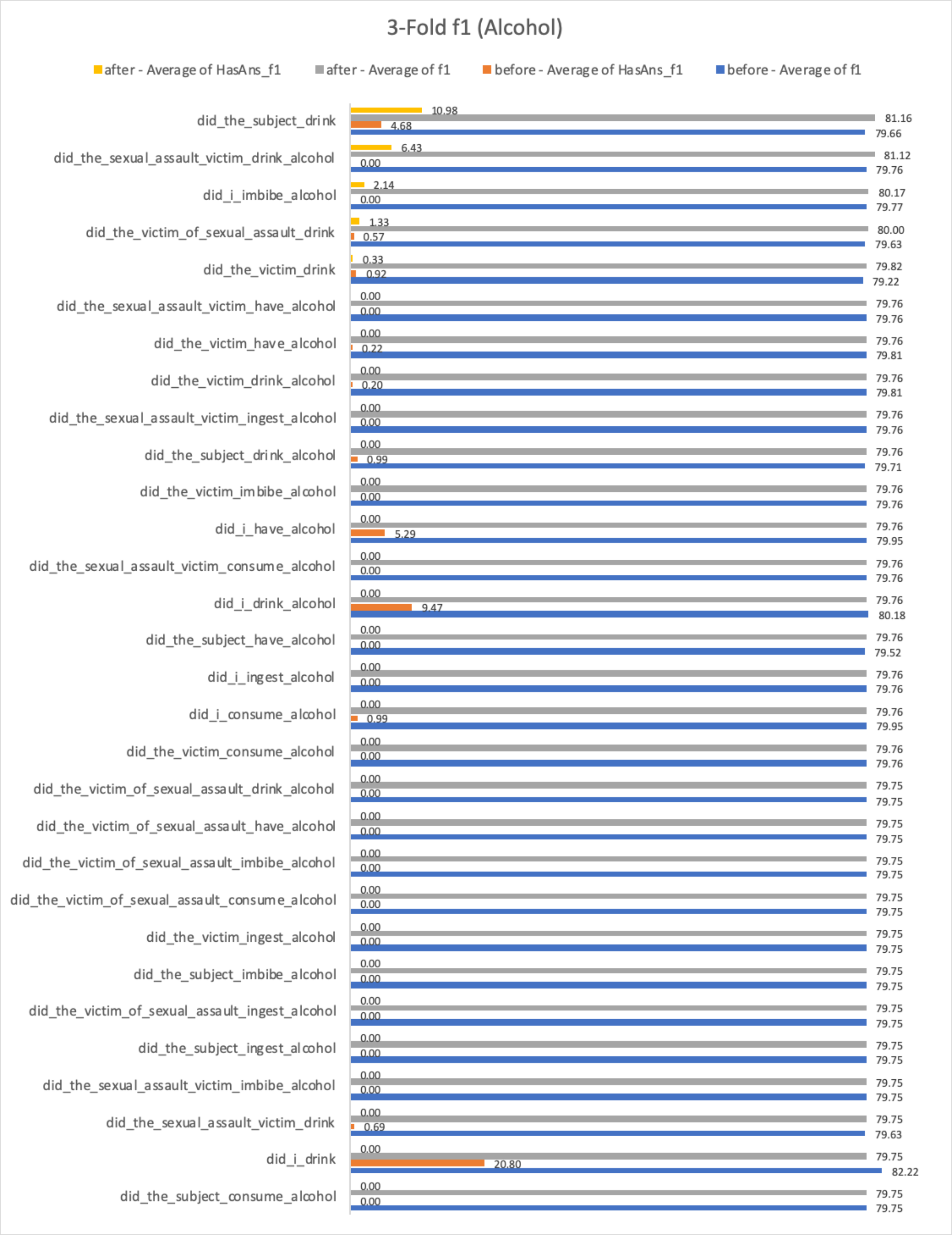
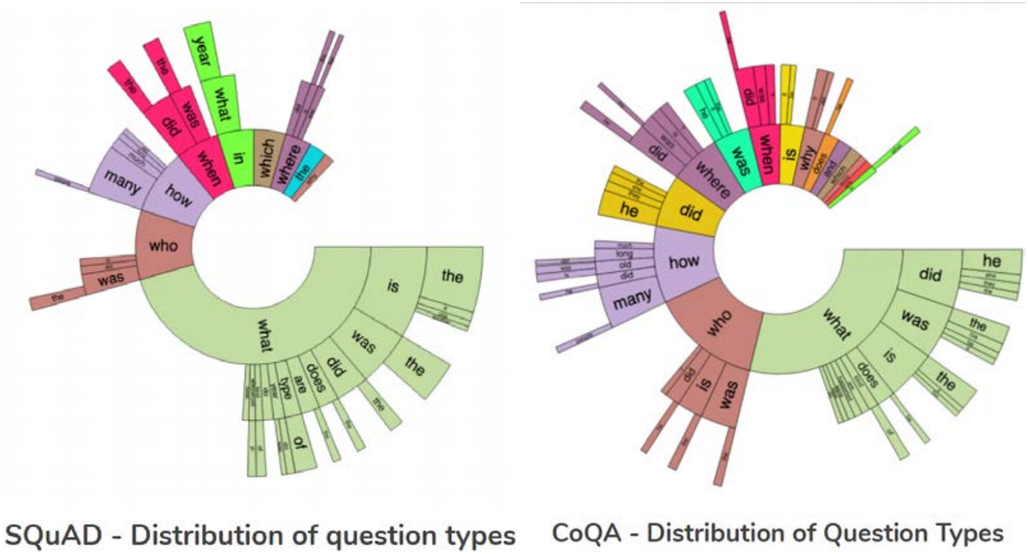


Figure 10: Output for the f1 scores of alcohol consumption questions. “f1” refers to the total f1 score, “HasAns_f1” refers to the f1 score when considering only questions with a defined true answer. “before” refers to using only the pre-trained Bert model with SQuAD data, “after” refers to using the pre-trained Bert mode that has been fine-tuned with our data.

Our highest f1 was 10.98%. For the most part, f1 scores went down after training, and often bottomed out at 0%. You can’t see this in the chart, but there was no consistency across folds and runs.

Clearly, something was wrong, but we had a few hypotheses for the cause. (1) There is a mismatch between the model and the data. Most of the comments had no references to drinking alcohol, and the ones that did had multiple references in the post, rather than just one, unlike the mention of age. This poses a problem for the BERT model, which can only predict a single span of text as the answer. This leads to (2): Although we can define multiple possible answers, this only applies to checking the results and isn’t useful when training. That means that the model is only trained on a limited subset of the alcohol drinking examples. (3) We had too few examples to train the alcohol questions on. (4) We missed another way to phrase our question.

When we ran the same test on drug data, we ended up with similar bad results. At this point, we were at a bit of a dead end with the BERT model. Based on hypotheses 1 and 2, we realized it didn’t have the flexibility to answer the questions we wanted it to answer. For alcohol drugs, we needed to answer “did” questions. SQuAD doesn’t contain these kinds of questions, so we would expect models trained on it won’t be able to answer them. Furthermore, the BERT model is unable to do any kind of transformation on the span it selects from the context, leaving it unable to answer “yes” or “no”, which we want for drugs and alcohol.



SQuAD - Distribution of question types CoQA - Distribution of Question Types

Figure 11: Difference in question types between the SQuAD format and the CoQA format.

CoQA, however, has “did” questions (Figure 11) and Microsoft’s SDNet model is able to transform text from the contexts (as in Q4 of Figure 2). This ability to transform text gives us another advantage: we can ask SDNet to directly output categories as answers to questions (with the caveat that this will require training to understand what the categories are).

With all these potential benefits from SDNet, we moved forward with training it on CoQA, and attempted to rerun our experiment on all four questions. We used the default SDNet configuration, with 30 epochs. Unfortunately, SDNet is also a large model (BERT is just a part of it) and requires heavy question preprocessing. While we were able to evaluate f1 scores for each question across all four data types, we were only able to fine tune on relationship data (and for just 5 epochs).

For age, SDNet (Figure 12) performed better than BERT's no-fine-tuning (Figure 5) for comments with an answer (normal_f1), but much worse at comments without an answer (no_f1). The best question was "What is my age?", which had a normal_f1 of 54.82%, but a total f1 of just 26.41%.

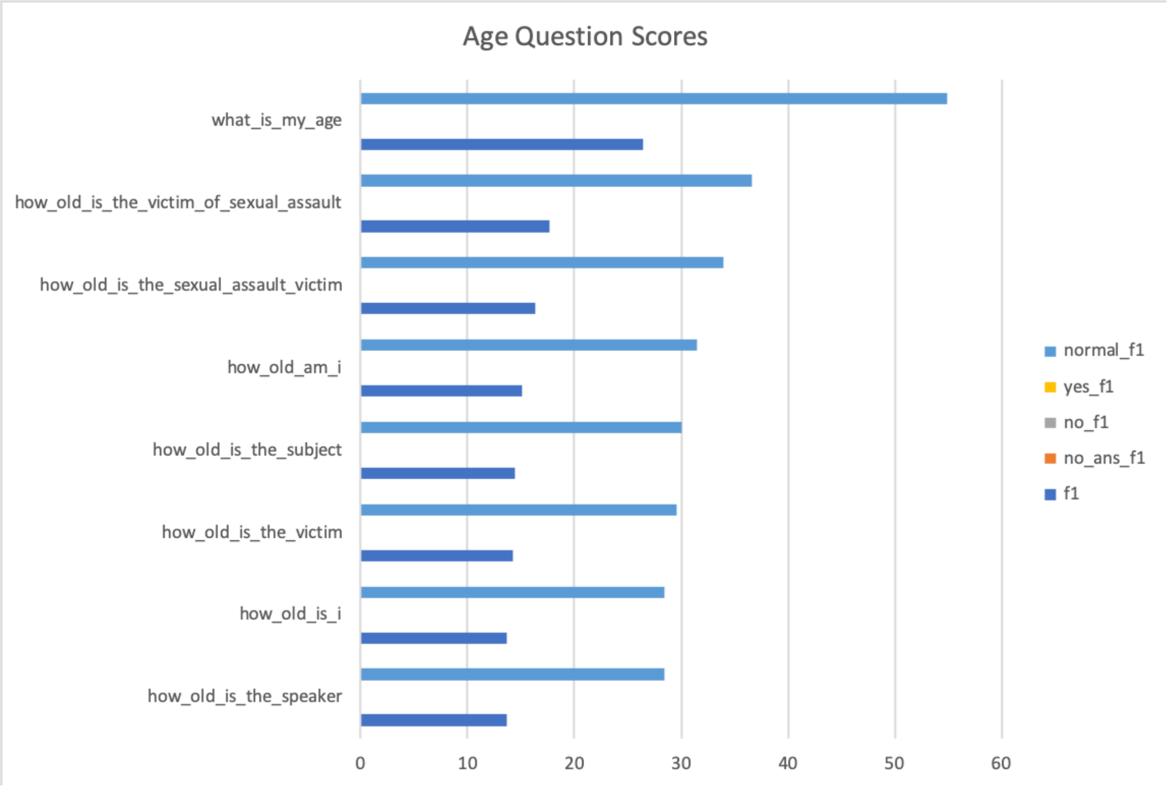


Figure 12: The f1 scores for age questions using SDNet trained on CoQA, sorted from high to low by "f1" (the blue bar). "normal_f1" is the score for regular questions. "yes_f1" is the score for entries with a truth label of "yes." "no_f1" is the score for entries with a truth label of "no." "no_ans_f1" is the entry for the score where there's no specified true answer value. "f1" is the total f1 score across all questions.

For alcohol, as we expected, SDNet (Figure 13) performed much better than BERT (Figure 10), both pre and post fine-tuning. The best question, "Did the victim of sexual assault drink?", gets an f1 of 85.12%, answers "no" with a 90.21% f1, and answers "yes" with a 65.06% f1. This is better than the baseline accuracy of 80%, based on the distribution of answers (Figure 4).

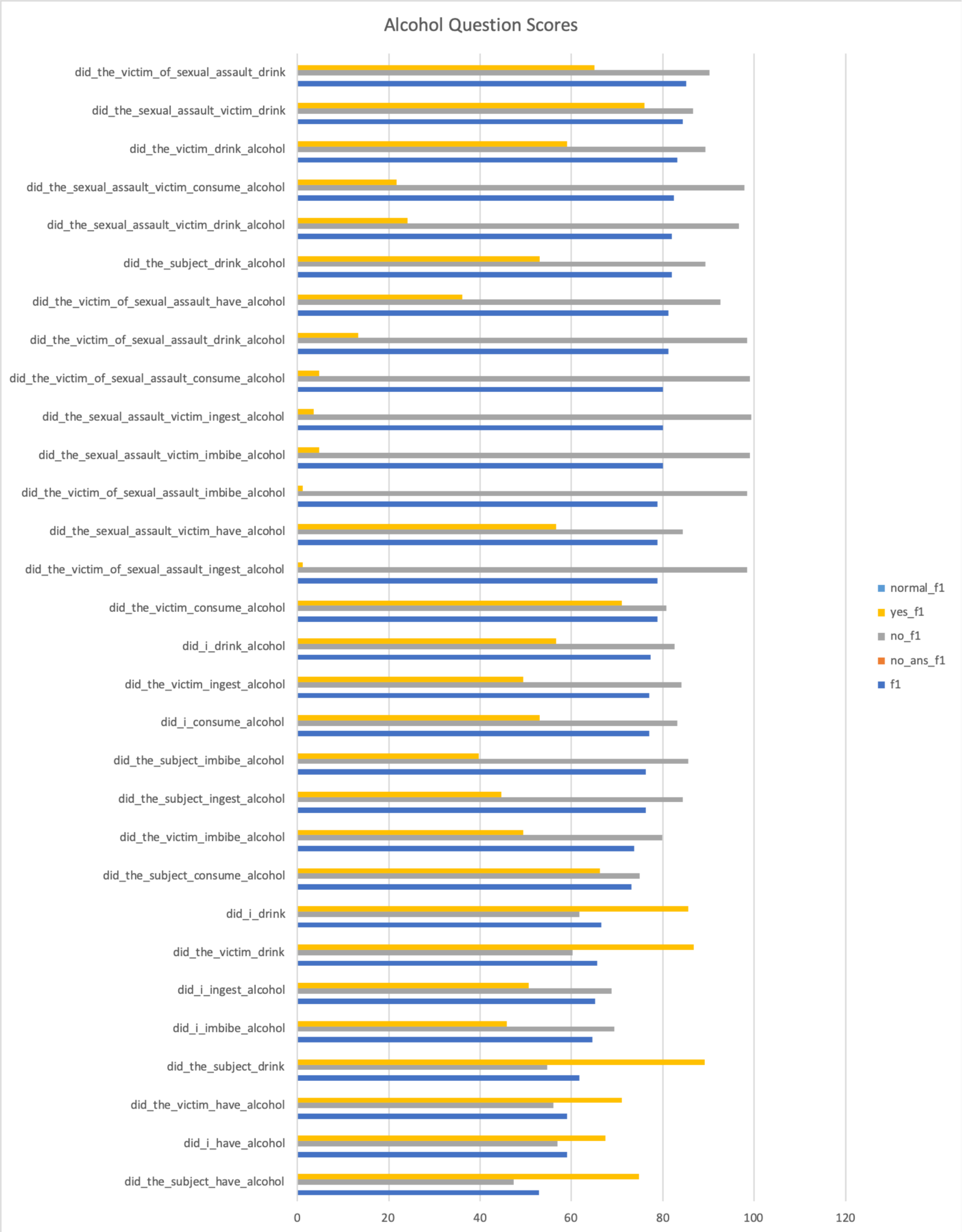


Figure 13: The f1 scores for alcohol questions using SDNet trained on CoQA, sorted from high to low by “f1” (the blue bar). “normal_f1” is the score for regular questions. “yes_f1” is the score for entries with a truth label of “yes.” “no_f1” is the score for entries with a truth label of “no.” “no_ans_f1” is the entry for the score where there’s no specified true answer value. “f1” is the total f1 score across all questions.

The drug questions give us some interesting results (Figure 14). It's still better than BERT, but no model is good at both no_f1 and yes_f1. Even though the best question, "Did the victim of sexual assault ingest drugs?" gets a total f1 of 92.68%, it has a yes_f1 of just 7.41%. And while no_f1 and overall f1 scores appear highly correlated, they have an inverse relationship to yes_f1 score. For perspective, the baseline of only guessing "no" is 93.41%. We believe that the data distribution, which is 93% no answers, contributes heavily towards this skew.

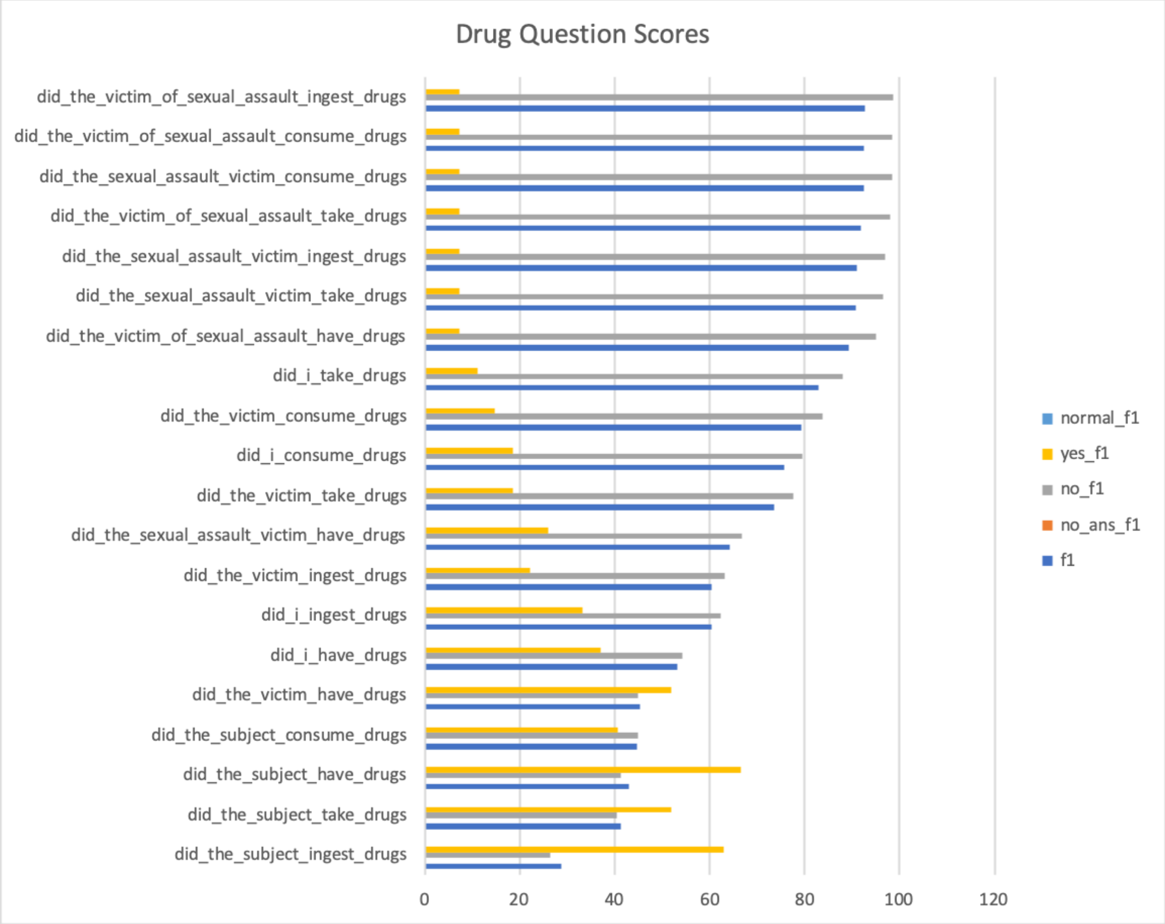


Figure 14: The f1 scores for drug questions using SDNet trained on CoQA, sorted from high to low by "f1" (the blue bar). "normal_f1" is the score for regular questions. "yes_f1" is the score for entries with a truth label of "yes." "no_f1" is the score for entries with a truth label of "no." "no_ans_f1" is the entry for the score where there's no specified true answer value. "f1" is the total f1 score across all questions.

For non-fine-tuned relationship questions (Figure 15), we have f1 scores of 0-0.5%. This is expected, since SDNet doesn't know we only want categories as answers. When we fine tune on 3 folds for 5 epochs, we end up with somewhat improved results (Figure 16). The new best question, "How did the victim know the suspect?", has an average f1 across folds of 1.99%. This is not significant enough to draw results from, but we hypothesize that training for more epochs might continue to improve the results.

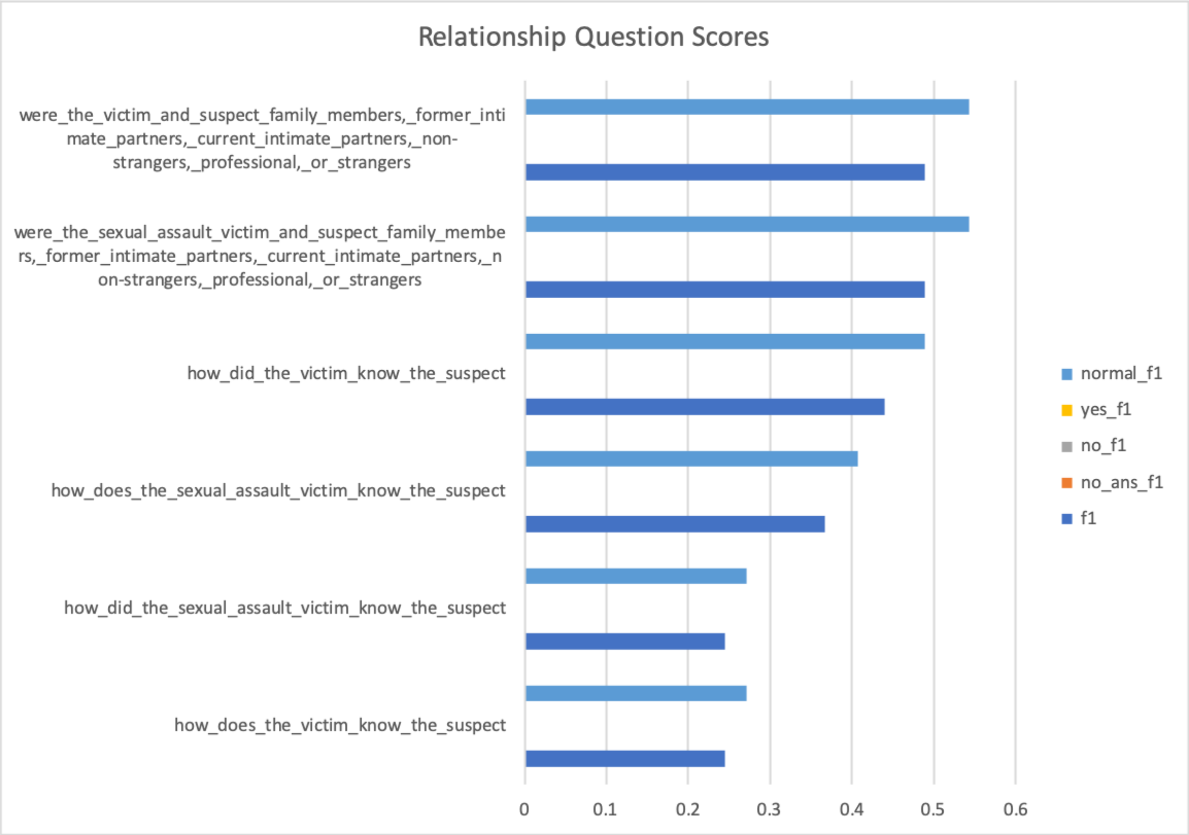


Figure 15: The f1 scores for relationship questions using SDNet trained on CoQA, sorted from high to low by “f1” (the blue bar). “normal_f1” is the score for regular questions. “yes_f1” is the score for entries with a truth label of “yes.” “no_f1” is the score for entries with a truth label of “no.” “no_ans_f1” is the entry for the score where there’s no specified true answer value. “f1” is the total f1 score across all questions.

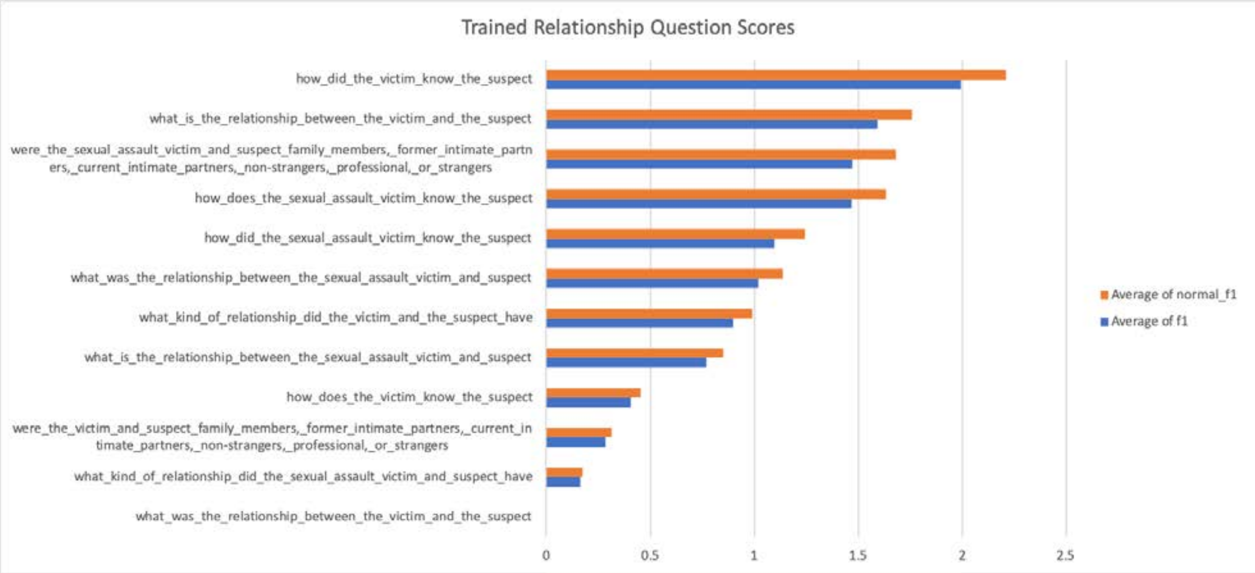
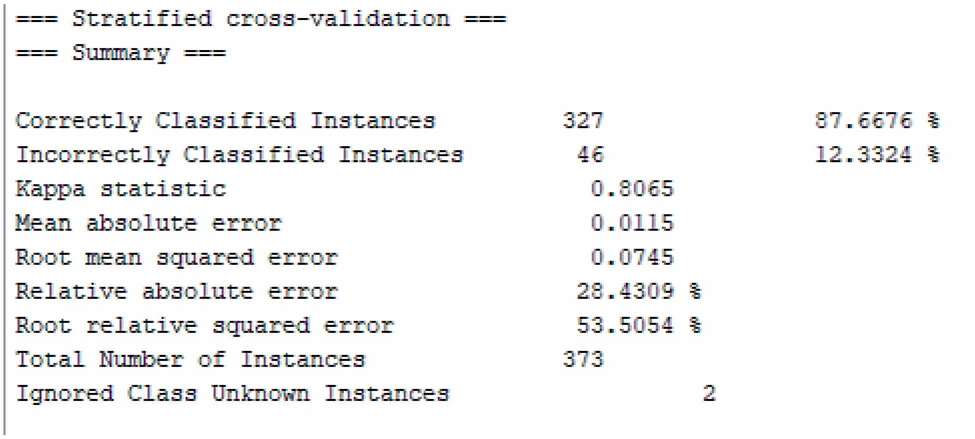


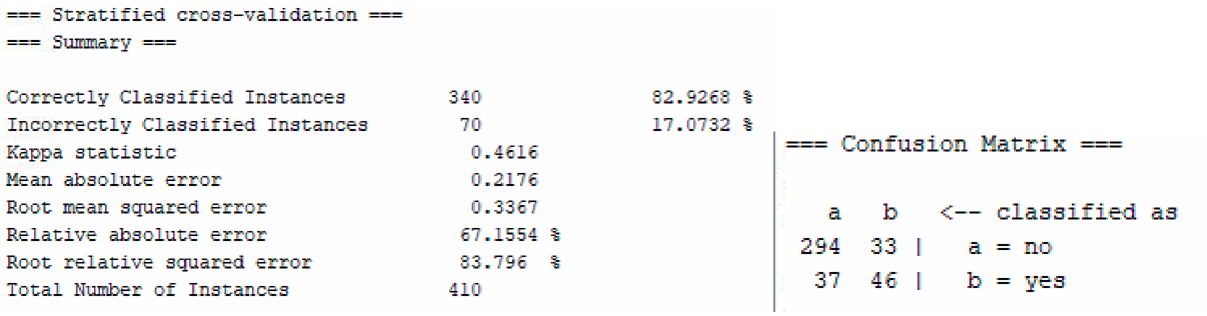
Figure 16: The average f1 scores for relationship questions using SDNet trained on CoQA and fine tuned on folds, sorted from high to low by “Average of f1” (the blue bar). “Average of normal_f1” is the score for regular questions. “Average f1” is the average total f1 score across all questions.



```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         327               87.6676 %
Incorrectly Classified Instances        46               12.3324 %
Kappa statistic                          0.8065
Mean absolute error                      0.0115
Root mean squared error                  0.0745
Relative absolute error                 28.4309 %
Root relative squared error             53.5054 %
Total Number of Instances              373
Ignored Class Unknown Instances                  2
```

Figure 17: The WEKA prediction summary and confusion matrix on age questions.

The WEKA prediction collation on age had an accuracy of 87.67%. The top three questions were “How old is the victim of sexual assault?”, “How old is I?”, and “How old am I?”. It is interesting that the importance level of a grammatically incorrect question is actually second in the set. We hypothesize this is associated with the fact that “when I was X” is the most popular manner of expressing the age in the contexts. It is quite interesting to see that adding a focus on sexual assault would make the last ranked question jump straight to the top and rank #1. The position of the last ranked question might be attributed to a high level of generalization causing the accuracy to drop. Of course the same caveats from above, when running WEKA on the BERT age question results, apply.

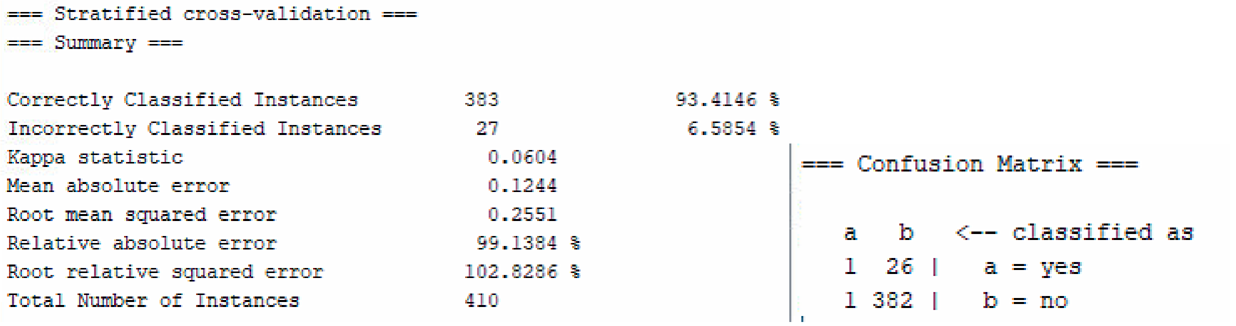


```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         340               82.9268 %
Incorrectly Classified Instances        70               17.0732 %
Kappa statistic                          0.4616
Mean absolute error                      0.2176
Root mean squared error                  0.3367
Relative absolute error                 67.1554 %
Root relative squared error             83.796  %
Total Number of Instances              410
```

```
=== Confusion Matrix ===

   a   b   <-- classified as
 294  33 |   a = no
  37  46 |   b = yes
```

Figure 18: The WEKA prediction summary and confusion matrix on alcohol questions.

The WEKA prediction on alcohol questions had an accuracy of 82.93%. The top three questions were “Did the sexual assault victim drink?”, “Did the victim of sexual assault drink?” and “Did the victim drink alcohol?” It’s interesting to note that the lowest scores were with questions that used the words “ingest” and “imbibe,” possibly due to the low common usage of those words.

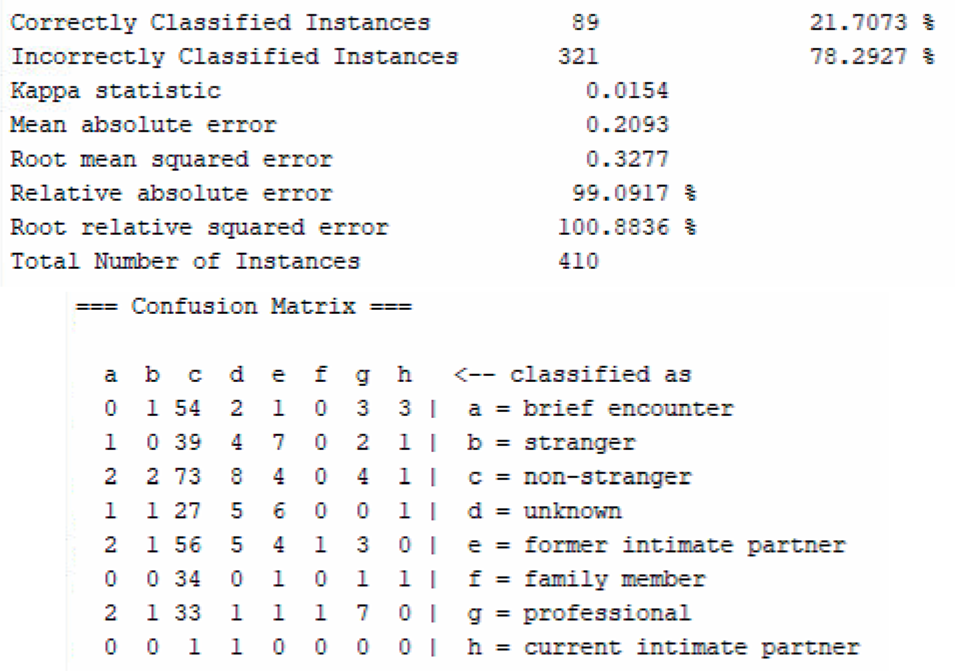


```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         383               93.4146 %
Incorrectly Classified Instances        27                6.5854 %
Kappa statistic                          0.0604
Mean absolute error                      0.1244
Root mean squared error                  0.2551
Relative absolute error                 99.1384 %
Root relative squared error            102.8286 %
Total Number of Instances              410
```

```
=== Confusion Matrix ===

   a   b   <-- classified as
   1  26 |   a = yes
   1 382 |   b = no
```

Figure 19: The WEKA prediction summary and confusion matrix on drug questions.

The WEKA prediction for drug questions had an accuracy of 93.41%, which matched the baseline prediction exactly. The top three questions were "Did the victim of sexual assault ingest drugs?", "Did the victim of sexual assault consume drugs", and "Did the victim of sexual assault take drugs?" It is worth noting here that the a higher specificity when mentioning the victim and using more particular words like "ingest" or "consume" with drugs have a significant impact on the performance as compared to more generic terms such as "have", or simply mentioning the sexual assault victim as victim.



```
Correctly Classified Instances          89               21.7073 %
Incorrectly Classified Instances       321               78.2927 %
Kappa statistic                          0.0154
Mean absolute error                      0.2093
Root mean squared error                  0.3277
Relative absolute error                 99.0917 %
Root relative squared error            100.8836 %
Total Number of Instances              410
```

```
=== Confusion Matrix ===

  a  b  c  d  e  f  g  h   <-- classified as
  0  1 54  2  1  0  3  3 |  a = brief encounter
  1  0 39  4  7  0  2  1 |  b = stranger
  2  2 73  8  4  0  4  1 |  c = non-stranger
  1  1 27  5  6  0  0  1 |  d = unknown
  2  1 56  5  4  1  3  0 |  e = former intimate partner
  0  0 34  0  1  0  1  1 |  f = family member
  2  1 33  1  1  1  7  0 |  g = professional
  0  0  1  1  0  0  0  0 |  h = current intimate partner
```

Figure 20: The WEKA prediction summary and confusion matrix on untrained relationship questions.

The WEKA prediction collation on untrained relationships surprisingly gave us an accuracy of 21.71%, compared to the top question f1 of <0.5%. We believe this is due to WEKA knowing the classes that we want (stranger, brief encounter, etc.) where SDNet doesn't have that information and resorts to just making guesses. The top three questions were "What was the relationship between the sexual assault victim and suspect?", "What was the relationship between the victim and the suspect?", and "What kind of relationship did the sexual assault victim and suspect have?" However, all three of those had a very low contribution score.

Workflow Summary

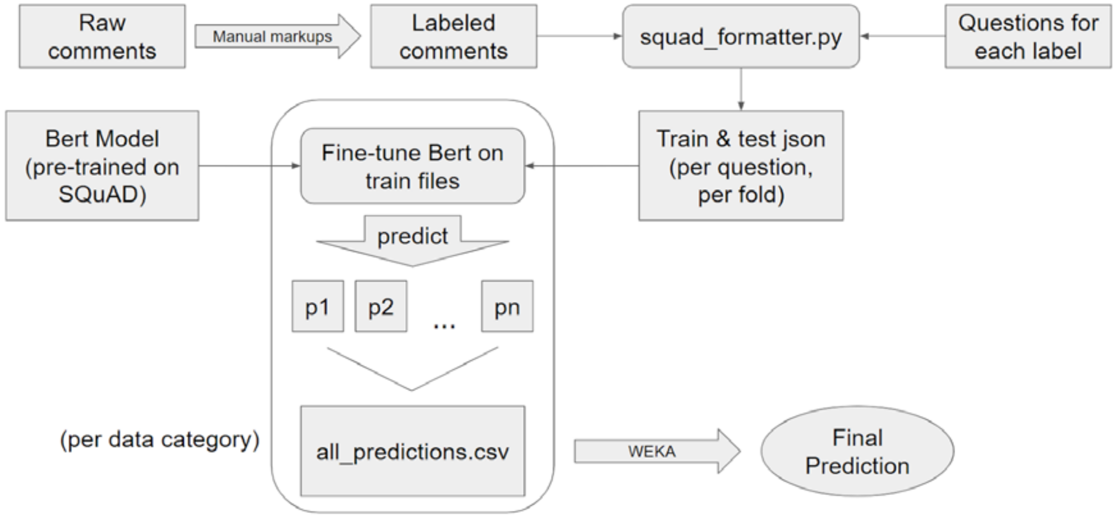


Figure 21: The workflow diagram for our project when using the Bert model.

Figures 21 and 22 detail our workflow models for working with Bert and with SDNet respectively. The key differences between the usage of the two models are the dataset (SQuAD vs Bert) as well as the format of the JSON files that each uses, but other than that, they are largely similar.

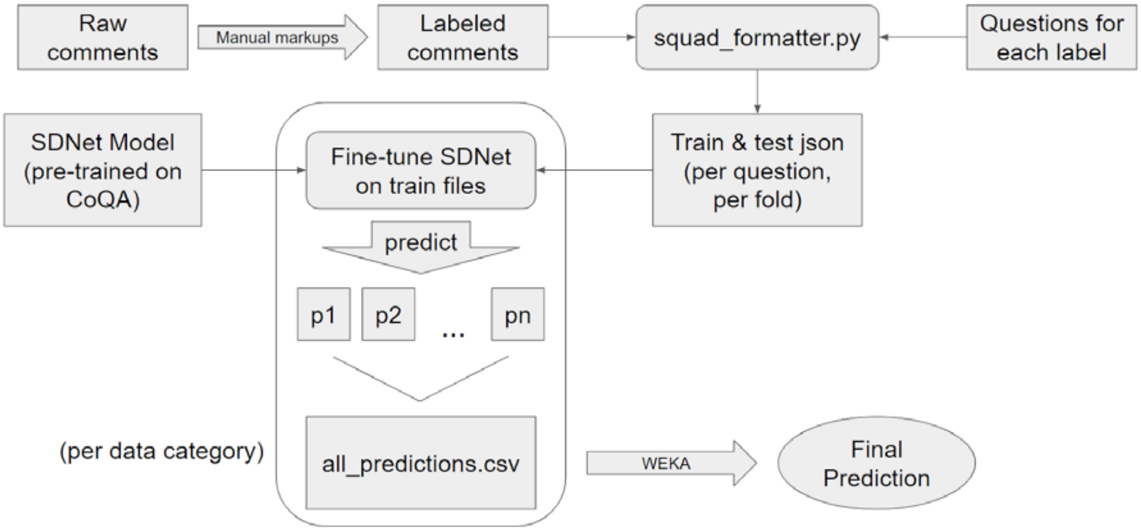


Figure 22: The workflow diagram for our project when using the SDNet model.

VI. Conclusion

We were able to achieve promising results for three of our four categories. For the age category, it's much more effective to use the Bert and SQuAD combination, which is good for quoting answers from text. For alcohol and drug consumption, it's more effective to use the SDNet and CoQA combination, which already has built in functionality for yes and no questions. Though the drug category had an f1 score just below its baseline when not using WEKA, we expect that a better distribution of the results would increase the prediction performance as it's comparable to the alcohol category. We were unable to achieve good results for the relationship category but have identified promising leads and paths forward. Finally, analysis of any narrative data from a third-person perspective is the real problem here. Although we used Reddit data as a substitute, is not the right fit because it is first-person (and therefore prone to biased views).

VII. Future Work

There are many avenues we would like to explore if we had the time. These include:

- Increase the epochs we fine tune SDNet with.
- Train SDNet for age, alcohol, and drugs, not just relationships.
- Explore using BERT-Large (in the BERT+SQuAD and SDNet model).
- Explore different hyperparameters for BERT.
- Explore different hyperparameters for SDNet.
- Ask leading questions, such as "Who is the victim?", in SDNet to establish context.
- Explore using a second classifier to map SDNet relationship question answers into the SAJI form's category.
- Switch to a third-person POV dataset.
- Expand to extract other kinds of information.
- Use only the top three questions to evaluate the performances.

VIII. Acknowledgements

First, we thank Professor Daniel Lopresti for offering the AI for Social Good course and allowing us the opportunity to work on this project, as well as providing guidance throughout the project's duration. We also wish to acknowledge Professor Sihong Xie for taking the time to speak with us and give us suggestions on how to proceed in our project. We also appreciate Google and Microsoft for freely publishing their Bert and SDNet models, respectively, which we utilized in this project. We also thank Stanford for freely publishing their SQuAD and CoQA datasets, which we trained off of. We thank the Waikato University in New Zealand for their WEKA tool and the help it gave us. Last, we thank AEquitas for providing us with the opportunity to work on this project, as we feel we learned a great deal from it.

IX. References

AEquitas, the Justice Management Institute, & the Urban Institute. (2018). *Sexual Assault Justice Initiative (SAJI) Draft Implementation Guide for Performance Management*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. Retrieved from https://arxiv.org/abs/1810.04805v1

Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques" - Morgan Kaufmann, Fourth Edition, 2016.

Greene, D., & Cunningham, P. (2006). Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering. *Proc. 23rd International Conference on Machine Learning (ICML'06)*, 377–384. Retrieved from http://mlg.ucd.ie/datasets/bbc.html

Rajpurkar, P., Jia, R., & Liang, P. (2018). Know What You Don't Know: Unanswerable Questions for SQuAD. *ArXiv:1806.03822 [Cs]*. Retrieved from http://arxiv.org/abs/1806.03822

Reddy, S., Chen, D., & Manning, C. D. (2018). CoQA: A Conversational Question Answering Challenge. *ArXiv:1808.07042 [Cs]*. Retrieved from http://arxiv.org/abs/1808.07042

Zhu, C., Zeng, M., & Huang, X. (2018). SDNet: Contextualized Attention-based Deep Network for Conversational Question Answering. *ArXiv:1812.03593 [Cs]*. Retrieved from http://arxiv.org/abs/1812.03593