Project Log Book

Meeting Notes

Meeting 1 - 7/10/20

Questions
- What kind of dataset should I be looking for?
- How does this years project differ from last years?
- How rigid is the domain?
- What sort of metrics are useful for checking model scores for this task? As answers are no longer binary

Could be general medical question answering - take a huge dataset
- Doesn't matter about specific discipline
- (very advanced objective) If we have enough time later we can think about how to answer different question types from the dataset and enrich it with more data

Expectations is that you go and find your own data
Noura will send me some papers to read

BioBERT - make the whole project Bio things
Don't try to find loads of word embeddings and compare against them
We can move the comparison to different things.
We can try normal bert and biobert

Do not try to train BERT from scratch
Pre-trained model - she will send me some BERT models and BIOBERT

Specific datasets, special models, embeddings for medical domain
What new things can you learn during this project (you should be able to use the transformers)

What's the state of the art in biomedical question answering, develop a proposal?
The literature survey lets you find existing solutions for the problem were solving
- They will probably be non-medical based
- You can recycle most of it later on
- As a number of words and so on the paper is the same. From l3 we expect them to use existing advanced tech and learned something new. For me to get over 70 or 80 this year, I have to develop the state of the art in this domain. You don't have to do it in all of medical, just in a niche way. Students who choose easy projects.
- I NEED to extend the state of the art.
- This is a research topic, when you speak to alexa, there is like an army of amazonians who engineer the questions and answers.

- It is very challenging, people do PHDs in this domain. I can expand as much as I want. The challenge is there and adding the medical aspect means it is more impactful.
- It is way more challenging than last time,

Meeting 2 - 13/10/20
Questions for Noura:
- Is it good practice to pick multiple datasets?

Approaches considered
- Switching out BERT for BioBERT in current QA sota to apply it to the biomedical domain
- Question entailment approaches which try to match a new question to a previously answered question
- There are multiple question types

Next steps:
- Build understanding of transfer learning

Meeting 3
Ideas (Feasibility):
- How good is GPT-2 at question answering.

- Proposal so far
    - Reformer is a more efficient transformer - like a transformer, it has an encoder and decoder mechanism
    - BERT includes the encoder mechanism of a transformer
    - Replace this mechanism with the encoder from the reformer

BERT has pre-trained weights. BioBERT begins with these weights, continues pre-training for 23 days on medical articles (Extracts from PubMed). Fine-tuned with small factoid dataset (BioASQ) ~ 1000 very high quality question answer pairs.

There aren't any pre-trained weights to initialise the reformer model to. Could pre-train on biomedical texts like BioBERT then finetune to question answering.

BioBERT used the same pre-training technique as Wiese et al. 2017, who in turn used the same training procedure as Weissenborn 2017.

GPT 2 vs GPT 3
Try and generate text in response to a question
See how coherent and relevant the answers are

There is an interface for text generator from transformers
Dataset is just papers or articles
We can fine-tune it with question answering
https://huggingface.co/mrm8488/GPT-2-finetuned-CORD19?text=My+name+is+Mariama%2C+my+favorite

Benefits of bert vs text generation transformers
Make a Figure categorising the transformers:
Gpt - text generation seq-2-seq
Bert uses masking

Then say which one were choosing
Computation cost is important (acceptable trade-off)

Fine-tuning a model is the best approach
Even loading a model could

Always focus on transformers as tools for qa
Bert does not generate text -> its good for multichoice answer
Depends on what we want to target -> if its multichoice

Base it on the most popular question type as then i will have a lot of materials for comparison and analysis.

If texts are long, then T5 is one of the best models to deal with long documents

T5, Reformer, XLNET (family of models), BERT based models, GPT
Always compare in context of question answering.

Meeting 4

- an "ideal answer" approach since this is more human-readable and the bioasq dataset has these answers available.
- May try and submit it to the BioASQ challenge 2020 if results are good enough

By next week:

- Write code to process BioASQ data (measure avg sequence length, num repetitions in the data, maybe topic modelling?)
- Add and refine my description of Electra and put it in the solution

Meeting 5

Debugged code on the NCC
Introduced a trained tokenizer

- Should I have more than one evaluation dataset for QA?

- Examine the BioASQ dataset:
    - Are there any data replications
    - What questions cover what part of data (bioasq)
    - What's the story of final paper (works with multi-topics (covid is one of them),
    -
    -
    - ovid, just a generic medical model) - think of how to sell the story.
    - What is the literature about every part (what is the challenging part about the data)

Aim for two weeks run
Randomly pick 14 million samples
Random sample

Take a random sample and check for repetition
Using it for training and validation

Meeting 6
Ultimate goal:

- Trained model that is ready to use
- Looked at the dataset
-
- What percentage data replicated:
- Replications can be in training but not test
- Trained model that can work
- A plan for the dataset
- Describe the electra model in the solution
- Put the lit review in a paper template and expand on that
- Talk about method and solution
- Download the evaluation form and populate it according to criteria
- Make sure introduction covers all aspects, method and solution

Submit a workshop paper

For next week:

- Try applying any version of checkpoints with the dataset and produce some results. Write the code like a framework if we decide to apply it on another data. Don't make it specific to bioasq. Finish fine-tuning code.
- Try and get some early results.
- Let here

Meeting:

Progress since last meeting:

- Created diagram of the pre-training and fine-tuning stage and wrote overview of solution
- Improved the stages of reading in bioasq (qa finetuning) dataset and converting this into more meaningful features
    - Now i'm able to read in squad and bioasq
    - For three question types, yesno, factoid and list, matching the format of bioasq

- Created an interface for interacting with models
- Contacted Kevin Clark about my pre-training results and imbalance between precision and recall.

BioASQ - See what other people have done from bioasq. Spend another week on trying to improve the performance without changing the data. Could try once class classifiers and see which prediction is the most confident. Give it another week of tweaking.
How do the current SOTA resolve these problems ^^^

Questions:

- Can we discuss improvements based on the diagram?
- Need to improve the results of my binary classification model - data is imbalance, tried cost-sensitive loss function which had a small effect.
- When i put my binary classification model into evaluation mode it just predicts yes all of the time, do you know what the reason could be?
- Output predictions for factoid and list questions are really bad

Term 2 Diary:

14th January 2021

- Backup trained epochs on durham fileserver to make more space on ncc
- Create json file from pubmed articles referenced in a bioasq training file

15th January 2021

- Checkpoint name and model size can now be passed to slurm scripts to train different model sizes from a specific checkpoint
    - Train from specific checkpoint small model e.g. sbatch pretrain_small.slurm -c "small_22_107883"
    - Train from specific checkpoint base model e.g. sbatch pretrain_base.slurm -c "base_22_107883"
    - Train from recent checkpoint e.g. sbatch pretrain_small.slurm -c "recent"
    - Train from recent checkpoint e.g. sbatch pretrain_base.slurm -c "recent"

16th January 2021

- Moved the checkpoints folder out of pre-train so it can also be accessed by fine tuning code
- Find instances of save_dir in the finetuning code and decide if they are referencing the pretrain checkpoints or the finetune checkpoints

Bioasq challenge plan:

- Create a separate project which is a clone of the previous project
- Get training files and put them in a directory where they can be passed to download_data python file to get the corresponding pubmed articles in a consistent format.

24th Jan

Remember to do this next time using the NCC - install nltk sentence splitter (better than spacy)
Pip install unidecode too
https://stackoverflow.com/questions/4867197/failed-loading-english-pickle-with-nltk-data-load

25th Jan

- Now able to load in the squad dataset - need to extend this to bioasq. We have a couple of issues with tokens that end in the middle of a token.
- Create a way of writing features to a file to make loading faster in future. Although, it's not strictly necessary.

- May be able to use squad_convert_examples_to_features.

For next week:
- Write example problem background (and submit this to lovelace colloquium).
- Fix bug in finetuning code
- Run finetuning

Submit to these conferences: Widening nlp and women in machine learning

30th January 2021
- Backed up more epochs on durham fileserver: this time at the location C:\Users\kgxj22\Documents\new_checkpoints_base\

Since last week:

Created finetuning code:
- Able to load finetuned checkpoints
- Works with squad but still need to get this working with bioasq data

Created evaluation code:
- Uses a finetuned model with test set and compares predicted answers against ground truth answers
- Created bioasq and squad metrics

This is for factoid questions, need to see how to produce a list of answers

Add extra features
Topic modelling concatenated to original embedding
Residual connections to the model
Attention between specific components of the model

- Create a flowchart of the final version of the model and come up with some improvements
- Make a flowchart and look at this together next week

Watch the lectures and then attend practical
Pytorch is able to be used in the practical. Coursework can be done in pytorch. Coursework is very linked to practicals.

2nd February 2021

- Latest estimate puts time for an epoch on base model at 98 hours (for approx)

Read in BioASQ data

- Split by question type (could possibly write to a file to avoid repeated work, or we can filter questions when we read them in for a particular model).
- There are about 3500 questions overall, so probably not very many.
- Make it so that evaluation can happen on multiple files (list of files)
- Change features for different question types (i.e. yes no needs labels whereas factoid needs start and end positions).

Use the ElectraForSequenceClassification model for tackling yes/no questions. We need to produce labels for them

Need to add the concept of a doc_stride, especially for electra small

Collate together the answers for yes no questions in evaluation stage. Take the most reported value (or sum together likelihoods for yes and no and see which is greater).

Yes no questions:

- Question id
- Question text
- Context (roughly a single sentence)
-

3rd February 2021

Need to have better clipping around the answer and a better way of matching the answer to the paragraph. So far, we have that roughly 12000 samples are produced, or which 7000+ are unanswerable due to lack of a suitable match in the dataset. We need to tackle this issue immediately.
We also need to parse sub-tokens.

- Can now produce results for yes/no questions
-

7th February 2021

- Now have a way of evaluating bioasq during a run - not sure what happens if we use squad
- Have a slurm script to test out - the command to run is
- sbatch finetune_small.slurm -pc "recent" -fc "empty" -qt "yesno" -d "bioasq"
- The empty in fc is important,

Tasks remaining:

- Get some results for factoid and list
- Enable mid token tokenization
- Tasks for inputting to the model

12th February 2021

Making squad compatible with the bioasq dataset again - wanting to check impossible questions etc.

13th February 2021

To pass in multiple question types e.g. factoid and list we can pass a string "factoid,list"

Tasks remaining:

- Make sure evaluation metrics can handle NoneType answers caused by impossible questions
- We need to find a way to either identify impossible answer predictions, or remove the ability to predict them entirely
- We need a better way of collating answers

1. **Enable Fine-tuning of model on Factoid and List questions together.**

When training the factoid / list model, we want to use all of the factoid examples, and all of the list examples together to train this model, since they work in a similar way.

- How can we do this so that we evaluate during fine-tuning too, we need to essentially evaluate on factoid AND list as we go, with the two separate test datasets.
- Either that, or we can create a broader "evaluate" method which takes a single test dataset data-loader and produces two categories of metrics for the two question types contained within it.

^^^ we now have a way of doing this, but the way we're presenting the metrics is ugly, and hard to interpret. Can we change this?

2. Create results table in project paper and populate with results.

3. **Improve creation of factoid predictions**

In the evaluate factoid method, we could be more efficient with how we pick the best 5 predictions and do this in a way which yields better results.

4. **Improve factoid matchmaking**

   When we're matchmaking for bioasq factoid and list questions, we need to do this in a better way so that fewer questions are deemed impossible.

5. **Interpret impossible predictions**

We feed our model the start and end positions of 0 and 0 (in features) to suggest that we think the question is impossible. If we get start and end positions of 0 and 0 we should interpret these as impossible - but is there a better way to do this?

^^^ we have now removed impossible predictions because they aren't really very useful.

6. We need a better way of compiling list question answers, as we can gauge some information about how many answers are required by some of the questions. Similarly, we need to know when an answer is not good enough to make the list.

13th February 2021

Tasks remaining:

- To pass in multiple question types e.g. factoid and list we can pass a string "factoid,list"

Commands to run:

Run fine-tuning on Squad, on most recent pre-trained checkpoint.
sbatch finetune_small.slurm -pc "recent" -fc "empty" -qt "factoid" -d "squad"

Run fine-tuning on bioasq, on most recent pre-trained checkpoint and list and fact
sbatch finetune_small.slurm -pc "recent" -fc "empty" -qt "factoid,list" -d "bioasq"

Ideas for Related Work

Discuss:

- GloVE, Elmo as initial contextual ideas - introduce transformed based language models and mention that we give an overview of transformers in our solution.
- Introduce BioBERT, and other Bio language models

SENTENCE SIMILARITY APPROACHES
http://www.lrec-conf.org/proceedings/lrec2020/pdf/2020.lrec-1.676.pdf
Contextualized Embeddings based Transformer Encoder for Sentence Similarity Modeling in Answer Selection Task
→ contrast with the BioASQ attempt at this that performed very well.

Experiments with Class Weights
Avg metrics for question type 'yesno' and dataset '/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B1_golden.json' are

{'accuracy': 0.68, 'precision': 0.68, 'recall': 1.0, 'f1_y': 0.8100000000000002, 'f1_n': 0.0, 'f1_ma': 0.4050000000000001}.
Avg metrics for question type 'yesno' and dataset '/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B2_golden.json' are {'accuracy': 0.75, 'precision': 0.75, 'recall': 1.0, 'f1_y': 0.8570000000000001, 'f1_n': 0.0, 'f1_ma': 0.428}.
Avg metrics for question type 'yesno' and dataset '/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B3_golden.json' are {'accuracy': 0.581, 'precision': 0.581, 'recall': 1.0, 'f1_y': 0.735, 'f1_n': 0.0, 'f1_ma': 0.367}.
Avg metrics for question type 'yesno' and dataset '/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B4_golden.json' are {'accuracy': 0.5380000000000001, 'precision': 0.5380000000000001, 'recall': 1.0, 'f1_y': 0.7000000000000001, 'f1_n': 0.0, 'f1_ma': 0.35000000000000003}.
Avg metrics for question type 'yesno' and dataset '/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B5_golden.json' are {'accuracy': 0.559, 'precision': 0.559, 'recall': 1.0, 'f1_y': 0.717, 'f1_n': 0.0, 'f1_ma': 0.35800000000000004}.

Mod	Training Metrics
With Class Weights [1., 4.]	Avg metrics for question type 'yesno' and dataset '/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B1_golden.json' are {'accuracy': 0.68, 'precision': 0.68, 'recall': 1.0, 'f1_y': 0.8100000000000002, 'f1_n': 0.0, 'f1_ma': 0.4050000000000001}. Avg metrics for question type 'yesno' and dataset '/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B2_golden.json' are {'accuracy': 0.75, 'precision': 0.75, 'recall': 1.0, 'f1_y': 0.8570000000000001, 'f1_n': 0.0, 'f1_ma': 0.428}. Avg metrics for question type 'yesno' and dataset '/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B3_golden.json' are {'accuracy': 0.581, 'precision': 0.581, 'recall': 1.0, 'f1_y': 0.735, 'f1_n': 0.0, 'f1_ma': 0.367}. Avg metrics for question type 'yesno' and dataset '/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B4_golden.json' are {'accuracy': 0.5380000000000001, 'precision': 0.5380000000000001, 'recall': 1.0, 'f1_y': 0.7000000000000001, 'f1_n': 0.0, 'f1_ma': 0.35000000000000003}. Avg metrics for question type 'yesno' and dataset '/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B5_golden.json' are {'accuracy': 0.559, 'precision': 0.559, 'recall': 1.0, 'f1_y': 0.717, 'f1_n': 0.0, 'f1_ma': 0.35800000000000004}.
With Class Weights [1., 0.2]	All dataset metrics {Path('/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B1_golden.json'): {'yesno': [{'accuracy': 0.68, 'precision': 0.68, 'recall': 1.0, 'f1_y': 0.81, 'f1_n': 0, 'f1_ma': 0.405}, {'accuracy': 0.68, 'precision': 0.68, 'recall': 1.0, 'f1_y':

	0.81, 'f1_n': 0, 'f1_ma': 0.405}, {'accuracy': 0.68, 'precision': 0.68, 'recall': 1.0, 'f1_y': 0.81, 'f1_n': 0, 'f1_ma': 0.405}, {'accuracy': 0.68, 'precision': 0.68, 'recall': 1.0, 'f1_y': 0.81, 'f1_n': 0, 'f1_ma': 0.405}, {'accuracy': 0.68, 'precision': 0.68, 'recall': 1.0, 'f1_y': 0.81, 'f1_n': 0, 'f1_ma': 0.405}, {'accuracy': 0.68, 'precision': 0.68, 'recall': 1.0, 'f1_y': 0.81, 'f1_n': 0, 'f1_ma': 0.405}]}, Path('/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B2_golden.json'): {'yesno': [{'accuracy': 0.75, 'precision': 0.75, 'recall': 1.0, 'f1_y': 0.857, 'f1_n': 0, 'f1_ma': 0.428}, {'accuracy': 0.75, 'precision': 0.75, 'recall': 1.0, 'f1_y': 0.857, 'f1_n': 0, 'f1_ma': 0.428}, {'accuracy': 0.75, 'precision': 0.75, 'recall': 1.0, 'f1_y': 0.857, 'f1_n': 0, 'f1_ma': 0.428}, {'accuracy': 0.75, 'precision': 0.75, 'recall': 1.0, 'f1_y': 0.857, 'f1_n': 0, 'f1_ma': 0.428}, {'accuracy': 0.75, 'precision': 0.75, 'recall': 1.0, 'f1_y': 0.857, 'f1_n': 0, 'f1_ma': 0.428}, {'accuracy': 0.75, 'precision': 0.75, 'recall': 1.0, 'f1_y': 0.857, 'f1_n': 0, 'f1_ma': 0.428}]}, Path('/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B3_golden.json'): {'yesno': [{'accuracy': 0.581, 'precision': 0.581, 'recall': 1.0, 'f1_y': 0.735, 'f1_n': 0, 'f1_ma': 0.367}, {'accuracy': 0.581, 'precision': 0.581, 'recall': 1.0, 'f1_y': 0.735, 'f1_n': 0, 'f1_ma': 0.367}, {'accuracy': 0.581, 'precision': 0.581, 'recall': 1.0, 'f1_y': 0.735, 'f1_n': 0, 'f1_ma': 0.367}, {'accuracy': 0.581, 'precision': 0.581, 'recall': 1.0, 'f1_y': 0.735, 'f1_n': 0, 'f1_ma': 0.367}, {'accuracy': 0.581, 'precision': 0.581, 'recall': 1.0, 'f1_y': 0.735, 'f1_n': 0, 'f1_ma': 0.367}, {'accuracy': 0.581, 'precision': 0.581, 'recall': 1.0, 'f1_y': 0.735, 'f1_n': 0, 'f1_ma': 0.367}]}, Path('/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B4_golden.json'): {'yesno': [{'accuracy': 0.538, 'precision': 0.538, 'recall': 1.0, 'f1_y': 0.7, 'f1_n': 0, 'f1_ma': 0.35}, {'accuracy': 0.538, 'precision': 0.538, 'recall': 1.0, 'f1_y': 0.7, 'f1_n': 0, 'f1_ma': 0.35}, {'accuracy': 0.538, 'precision': 0.538, 'recall': 1.0, 'f1_y': 0.7, 'f1_n': 0, 'f1_ma': 0.35}, {'accuracy': 0.538, 'precision': 0.538, 'recall': 1.0, 'f1_y': 0.7, 'f1_n': 0, 'f1_ma': 0.35}, {'accuracy': 0.538, 'precision': 0.538, 'recall': 1.0, 'f1_y': 0.7, 'f1_n': 0, 'f1_ma': 0.35}, {'accuracy': 0.538, 'precision': 0.538, 'recall': 1.0, 'f1_y': 0.7, 'f1_n': 0, 'f1_ma': 0.35}]}, Path('/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B5_golden.json'): {'yesno': [{'accuracy': 0.559, 'precision': 0.559, 'recall': 1.0, 'f1_y': 0.717, 'f1_n': 0, 'f1_ma': 0.358}, {'accuracy': 0.559, 'precision': 0.559, 'recall': 1.0, 'f1_y': 0.717, 'f1_n': 0, 'f1_ma': 0.358}, {'accuracy': 0.559, 'precision': 0.559, 'recall': 1.0, 'f1_y': 0.717, 'f1_n': 0, 'f1_ma': 0.358}, {'accuracy': 0.559, 'precision': 0.559, 'recall': 1.0, 'f1_y': 0.717, 'f1_n': 0, 'f1_ma': 0.358}, {'accuracy': 0.559, 'precision': 0.559, 'recall': 1.0, 'f1_y': 0.717, 'f1_n': 0, 'f1_ma': 0.358}, {'accuracy': 0.559, 'precision': 0.559, 'recall': 1.0, 'f1_y': 0.717, 'f1_n': 0, 'f1_ma': 0.358}]}} Avg metrics for question type 'yesno' and dataset '/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B1_golden.json' are {'accuracy': 0.68, 'precision': 0.68, 'recall': 1.0, 'f1_y': 0.8100000000000002, 'f1_n': 0.0, 'f1_ma': 0.4050000000000001}. Avg metrics for question type 'yesno' and dataset '/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B2_golden.json' are {'accuracy': 0.75, 'precision': 0.75, 'recall': 1.0, 'f1_y': 0.8570000000000001, 'f1_n': 0.0, 'f1_ma': 0.428}. Avg metrics for question type 'yesno' and dataset '/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B3_golden.json

	n' are {'accuracy': 0.581, 'precision': 0.581, 'recall': 1.0, 'f1_y': 0.735, 'f1_n': 0.0, 'f1_ma': 0.367}. Avg metrics for question type 'yesno' and dataset '/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B4_golden.json' are {'accuracy': 0.5380000000000001, 'precision': 0.5380000000000001, 'recall': 1.0, 'f1_y': 0.7000000000000001, 'f1_n': 0.0, 'f1_ma': 0.35000000000000003}. Avg metrics for question type 'yesno' and dataset '/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B5_golden.json' are {'accuracy': 0.559, 'precision': 0.559, 'recall': 1.0, 'f1_y': 0.717, 'f1_n': 0.0, 'f1_ma': 0.35800000000000004}.
With Class Weights [4., 1.0]	All dataset metrics {Path('/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B1_golden.json'): {'yesno': [{'accuracy': 0.68, 'precision': 0.68, 'recall': 1.0, 'f1_y': 0.81, 'f1_n': 0, 'f1_ma': 0.405}, {'accuracy': 0.68, 'precision': 0.68, 'recall': 1.0, 'f1_y': 0.81, 'f1_n': 0, 'f1_ma': 0.405}, {'accuracy': 0.68, 'precision': 0.68, 'recall': 1.0, 'f1_y': 0.81, 'f1_n': 0, 'f1_ma': 0.405}, {'accuracy': 0.68, 'precision': 0.68, 'recall': 1.0, 'f1_y': 0.81, 'f1_n': 0, 'f1_ma': 0.405}, {'accuracy': 0.68, 'precision': 0.68, 'recall': 1.0, 'f1_y': 0.81, 'f1_n': 0, 'f1_ma': 0.405}, {'accuracy': 0.68, 'precision': 0.68, 'recall': 1.0, 'f1_y': 0.81, 'f1_n': 0, 'f1_ma': 0.405}]}, Path('/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B2_golden.json'): {'yesno': [{'accuracy': 0.75, 'precision': 0.75, 'recall': 1.0, 'f1_y': 0.857, 'f1_n': 0, 'f1_ma': 0.428}, {'accuracy': 0.75, 'precision': 0.75, 'recall': 1.0, 'f1_y': 0.857, 'f1_n': 0, 'f1_ma': 0.428}, {'accuracy': 0.75, 'precision': 0.75, 'recall': 1.0, 'f1_y': 0.857, 'f1_n': 0, 'f1_ma': 0.428}, {'accuracy': 0.75, 'precision': 0.75, 'recall': 1.0, 'f1_y': 0.857, 'f1_n': 0, 'f1_ma': 0.428}, {'accuracy': 0.75, 'precision': 0.75, 'recall': 1.0, 'f1_y': 0.857, 'f1_n': 0, 'f1_ma': 0.428}, {'accuracy': 0.75, 'precision': 0.75, 'recall': 1.0, 'f1_y': 0.857, 'f1_n': 0, 'f1_ma': 0.428}]}, Path('/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B3_golden.json'): {'yesno': [{'accuracy': 0.581, 'precision': 0.581, 'recall': 1.0, 'f1_y': 0.735, 'f1_n': 0, 'f1_ma': 0.367}, {'accuracy': 0.581, 'precision': 0.581, 'recall': 1.0, 'f1_y': 0.735, 'f1_n': 0, 'f1_ma': 0.367}, {'accuracy': 0.581, 'precision': 0.581, 'recall': 1.0, 'f1_y': 0.735, 'f1_n': 0, 'f1_ma': 0.367}, {'accuracy': 0.581, 'precision': 0.581, 'recall': 1.0, 'f1_y': 0.735, 'f1_n': 0, 'f1_ma': 0.367}, {'accuracy': 0.581, 'precision': 0.581, 'recall': 1.0, 'f1_y': 0.735, 'f1_n': 0, 'f1_ma': 0.367}, {'accuracy': 0.581, 'precision': 0.581, 'recall': 1.0, 'f1_y': 0.735, 'f1_n': 0, 'f1_ma': 0.367}]}, Path('/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B4_golden.json'): {'yesno': [{'accuracy': 0.538, 'precision': 0.538, 'recall': 1.0, 'f1_y': 0.7, 'f1_n': 0, 'f1_ma': 0.35}, {'accuracy': 0.538, 'precision': 0.538, 'recall': 1.0, 'f1_y': 0.7, 'f1_n': 0, 'f1_ma': 0.35}, {'accuracy': 0.538, 'precision': 0.538, 'recall': 1.0, 'f1_y': 0.7, 'f1_n': 0, 'f1_ma': 0.35}, {'accuracy': 0.538, 'precision': 0.538, 'recall': 1.0, 'f1_y': 0.7, 'f1_n': 0, 'f1_ma': 0.35}, {'accuracy': 0.538, 'precision': 0.538, 'recall': 1.0, 'f1_y': 0.7, 'f1_n': 0, 'f1_ma': 0.35}, {'accuracy': 0.538, 'precision': 0.538, 'recall': 1.0, 'f1_y': 0.7, 'f1_n': 0, 'f1_ma': 0.35}]}, Path('/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B5_golde

	n.json'): {'yesno': [{'accuracy': 0.559, 'precision': 0.559, 'recall': 1.0, 'f1_y': 0.717, 'f1_n': 0, 'f1_ma': 0.358}, {'accuracy': 0.559, 'precision': 0.559, 'recall': 1.0, 'f1_y': 0.717, 'f1_n': 0, 'f1_ma': 0.358}, {'accuracy': 0.559, 'precision': 0.559, 'recall': 1.0, 'f1_y': 0.717, 'f1_n': 0, 'f1_ma': 0.358}, {'accuracy': 0.559, 'precision': 0.559, 'recall': 1.0, 'f1_y': 0.717, 'f1_n': 0, 'f1_ma': 0.358}, {'accuracy': 0.559, 'precision': 0.559, 'recall': 1.0, 'f1_y': 0.717, 'f1_n': 0, 'f1_ma': 0.358}, {'accuracy': 0.559, 'precision': 0.559, 'recall': 1.0, 'f1_y': 0.717, 'f1_n': 0, 'f1_ma': 0.358}]}} Avg metrics for question type 'yesno' and dataset '/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B1_golden.json' are {'accuracy': 0.68, 'precision': 0.68, 'recall': 1.0, 'f1_y': 0.8100000000000002, 'f1_n': 0.0, 'f1_ma': 0.4050000000000001}. Avg metrics for question type 'yesno' and dataset '/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B2_golden.json' are {'accuracy': 0.75, 'precision': 0.75, 'recall': 1.0, 'f1_y': 0.8570000000000001, 'f1_n': 0.0, 'f1_ma': 0.428}. Avg metrics for question type 'yesno' and dataset '/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B3_golden.json' are {'accuracy': 0.581, 'precision': 0.581, 'recall': 1.0, 'f1_y': 0.735, 'f1_n': 0.0, 'f1_ma': 0.367}. Avg metrics for question type 'yesno' and dataset '/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B4_golden.json' are {'accuracy': 0.5380000000000001, 'precision': 0.5380000000000001, 'recall': 1.0, 'f1_y': 0.7000000000000001, 'f1_n': 0.0, 'f1_ma': 0.35000000000000003}. Avg metrics for question type 'yesno' and dataset '/home2/kgxj22/BiomedicalQA/Code/datasets/bioasq/raw_data/8B5_golden.json' are {'accuracy': 0.559, 'precision': 0.559, 'recall': 1.0, 'f1_y': 0.717, 'f1_n': 0.0, 'f1_ma': 0.35800000000000004}.

22nd February 2021

Discovered that binary classification models are not updated

23rd February 2021

Getting the most advanced checkpoints and checking that training is working properly.

- After 2 epochs, the base model is training properly
- After 11 epochs, the small model is still learning

Questions for NLP coursework session

- Will we be marked based on performance / model choice and justification. Like using an LSTM (sequential) or bi-lstm etc.
- For example, if I enrich the features with topic modelling stuff will i get more marks or what?

25th February 2021

- BoolQ dataset contains

28th January 2021

97500	Gettting bioasq stats for training 8b ------- COLLATING TRAIN-SET METRICS ------- Across all question types, there are 2466 questions and 30642 examples ------- LIST METRICS ------- Average questiion and context length: (64.94, 261.09) Created 11968 examples from 644 questions list-type questions account for 26.12% of total questions and 39.06% of total examples - 432 examples were skipped. ------- FACTOID METRICS ------- Average questiion and context length: (61.42, 230.52) Created 6698 examples from 941 questions factoid-type questions account for 38.16% of total questions and 21.86% of total examples - 435 examples were skipped. ------- YESNO METRICS ------- Average question and context length: (61.24, 201.08) Created 11976 examples from 881 questions yesno-type questions account for 35.73% of total questions and 39.08% of total examples - Positive Instances: 704 questions (79.91%), 10285 examples (85.88%) - Negative Instances: 177 questions (20.09%), 1691 examples (14.12%) Yes no weights are (6.082199881726789, 1.0) ------- COLLATING FEATURE METRICS ------- Created 11976 features from 11976 examples 0 examples were skipped in total due to the following errors: - 0 errors (0%): Ground truth answer does not match joined token answer - 0 errors (0%): Length of pre-tokenized context does not match length of

	original context - 0 errors (0%): Map returned by sub-tokenize was empty ------- COLLATING FEATURE METRICS ------- Created 68406 features from 11536 examples 78 examples were skipped in total due to the following errors: - 56 errors (71.79%): Ground truth answer does not match joined token answer - 22 errors (28.21%): Length of pre-tokenized context does not match length of original context - 0 errors (0.0%): Map returned by sub-tokenize was empty ------- COLLATING FEATURE METRICS ------- Created 36950 features from 6263 examples 65 examples were skipped in total due to the following errors: - 61 errors (93.85%): Ground truth answer does not match joined token answer - 4 errors (6.15%): Length of pre-tokenized context does not match length of original context - 0 errors (0.0%): Map returned by sub-tokenize was empty Created 117332 train features of length 128. ------- COLLATING TEST-SET METRICS ------- Across all question types, there are 378 questions and 3294 examples ------- LIST METRICS ------- Average questiion and context length: (48.96, 303.34) Created 1222 examples from 75 questions list-type questions account for 19.84% of total questions and 37.1% of total examples - 69 examples were skipped. ------- FACTOID METRICS ------- Average questiion and context length: (58.01, 307.74) Created 810 examples from 151 questions factoid-type questions account for 39.95% of total questions and 24.59% of total examples - 47 examples were skipped. ------- YESNO METRICS ------- Average questiion and context length: (58.51, 190.72) Created 1262 examples from 152 questions yesno-type questions account for 40.21% of total questions and 38.31% of total examples - Positive Instances: 95 questions (62.5%), 892 examples (70.68%) - Negative Instances: 57 questions (37.5%), 370 examples (29.32%) ------- COLLATING FEATURE METRICS ------- Created 1262 features from 1262 examples 0 examples were skipped in total due to the following errors: - 0 errors (0%): Ground truth answer does not match joined token answer

	- 0 errors (0%): Length of pre-tokenized context does not match length of original context - 0 errors (0%): Map returned by sub-tokenize was empty Created 1262 test features of length 128 from yesno questions. ------- COLLATING FEATURE METRICS ------- Created 6560 features from 1153 examples 55 examples were skipped in total due to the following errors: - 26 errors (47.27%): Ground truth answer does not match joined token answer - 29 errors (52.73%): Length of pre-tokenized context does not match length of original context - 0 errors (0.0%): Map returned by sub-tokenize was empty Created 6560 test features of length 128 from list questions. ------- COLLATING FEATURE METRICS ------- Created 4220 features from 763 examples 57 examples were skipped in total due to the following errors: - 30 errors (52.63%): Ground truth answer does not match joined token answer - 27 errors (47.37%): Length of pre-tokenized context does not match length of original context - 0 errors (0.0%): Map returned by sub-tokenize was empty Created 4220 test features of length 128 from factoid questions.
97501	Train on 9b and measure on combined test sets. Training empty check point for 30 epochs to see where plateau happens and reduce learning rate.
97526	Train on bioasq, evaluate on all 8b training sets separately. Learning rate is 1x10-4. {'yesno': {'avg': {'accuracy': 0.8258064516129039, 'precision': 0.8382903225806446, 'recall': 0.9524516129032254, 'f1_y': 0.8862258064516128, 'f1_n': 0.5759354838709677, 'f1_ma': 0.7310967741935486}, 'best': {'accuracy': 0.92, 'precision': 0.941, 'recall': 0.941, 'f1_y': 0.941, 'f1_n': 0.875, 'f1_ma': 0.908}}} {'yesno': {'avg': {'accuracy': 0.8923548387096778, 'precision': 0.8850322580645157, 'recall': 1.0, 'f1_y': 0.9364193548387094, 'f1_n': 0.6250322580645161, 'f1_ma': 0.7808064516129032}, 'best': {'accuracy': 0.972, 'precision': 0.964, 'recall': 1.0, 'f1_y': 0.982, 'f1_n': 0.941, 'f1_ma': 0.962}}} {'yesno': {'avg': {'accuracy': 0.8472580645161291, 'precision': 0.8314193548387092, 'recall': 0.9964193548387097, 'f1_y': 0.8971290322580643, 'f1_n': 0.6704193548387101, 'f1_ma': 0.7833548387096776}, 'best': {'accuracy': 0.968, 'precision': 0.947, 'recall': 1.0, 'f1_y': 0.973, 'f1_n': 0.96, 'f1_ma': 0.966}}} {'yesno': {'avg': {'accuracy': 0.8324193548387097, 'precision':

	0.8200000000000001, 'recall': 0.9954193548387097, 'f1_y': 0.8852903225806452, 'f1_n': 0.6663870967741936, 'f1_ma': 0.7759032258064517}, 'best': {'accuracy': 1.0, 'precision': 1.0, 'recall': 1.0, 'f1_y': 1.0, 'f1_n': 1.0, 'f1_ma': 1.0}}} {'yesno': {'avg': {'accuracy': 0.843741935483871, 'precision': 0.8265483870967738, 'recall': 0.9982903225806451, 'f1_y': 0.893225806451613, 'f1_n': 0.6752258064516129, 'f1_ma': 0.7844193548387096}, 'best': {'accuracy': 0.971, 'precision': 0.95, 'recall': 1.0, 'f1_y': 0.974, 'f1_n': 0.965, 'f1_ma': 0.97}}}
97527	Train on boolq, evaluate on all 8b training sets separately.
97535	Continuation - BioASQ with learning rate 3 x 10-4 {'yesno': {'avg': {'accuracy': 0.8361290322580641, 'precision': 0.8158387096774199, 'recall': 0.9886129032258064, 'f1_y': 0.8925483870967743, 'f1_n': 0.6396774193548385, 'f1_ma': 0.7660967741935489}, 'best': {'accuracy': 0.88, 'precision': 0.889, 'recall': 0.941, 'f1_y': 0.914, 'f1_n': 0.8, 'f1_ma': 0.857}}} {'yesno': {'avg': {'accuracy': 0.8575483870967742, 'precision': 0.8434516129032252, 'recall': 0.9976129032258065, 'f1_y': 0.9134193548387097, 'f1_n': 0.5827741935483872, 'f1_ma': 0.7480645161290326}, 'best': {'accuracy': 0.889, 'precision': 0.871, 'recall': 1.0, 'f1_y': 0.931, 'f1_n': 0.715, 'f1_ma': 0.823}}} {'yesno': {'avg': {'accuracy': 0.8717741935483868, 'precision': 0.8395806451612897, 'recall': 0.9874193548387097, 'f1_y': 0.9039032258064513, 'f1_n': 0.7865161290322584, 'f1_ma': 0.8451290322580642}, 'best': {'accuracy': 0.935, 'precision': 0.9, 'recall': 1.0, 'f1_y': 0.947, 'f1_n': 0.917, 'f1_ma': 0.932}}} {'yesno': {'avg': {'accuracy': 0.7916129032258065, 'precision': 0.7297419354838707, 'recall': 0.9977096774193549, 'f1_y': 0.8410967741935484, 'f1_n': 0.6818064516129034, 'f1_ma': 0.7616451612903226}, 'best': {'accuracy': 0.846, 'precision': 0.778, 'recall': 1.0, 'f1_y': 0.875, 'f1_n': 0.8, 'f1_ma': 0.838}}} {'yesno': {'avg': {'accuracy': 0.8853225806451613, 'precision': 0.853483870967742, 'recall': 0.9897741935483871, 'f1_y': 0.9122258064516128, 'f1_n': 0.8132903225806449, 'f1_ma': 0.8628709677419352}, 'best': {'accuracy': 0.971, 'precision': 0.95, 'recall': 1.0, 'f1_y': 0.974, 'f1_n': 0.965, 'f1_ma': 0.97}}}
97537	Continuation - BoolQ with learning rate 3 x 10-4 {'yesno': {'avg': {'accuracy': 0.8920645161290326, 'precision':

	0.8818387096774193, 'recall': 0.9809677419354834, 'f1_y': 0.9239999999999999, 'f1_n': 0.7887096774193553, 'f1_ma': 0.8564193548387098}, 'best': {'accuracy': 0.978, 'precision': 0.982, 'recall': 0.982, 'f1_y': 0.982, 'f1_n': 0.972, 'f1_ma': 0.977}}}

4th March

To-do list:

- ~~Fix MRR metric~~
- ~~Make predictions on test data, regardless of whether we have the answer. We don't want to search for the answer in the paragraph.~~
- ~~Remove the concept of short and long context since we only use 1 now anyway.~~
- We normalised our answers, but we may need to undo this for the final prediction.
- We need a better way of collating the answers for list questions - for factoid, we just take the top 5
- Use contains_k function for list questions
- See if we can get the number of expected answers from the list.
- We could try a weighted answer selection
- Add probability thresholding approach
- Find good values for probability thresholding
- ~~we need to make sure we don't keep duplicates in our list~~
- ~~Find out what value of k to use for bioasq factoid and list questions - k=5 and k=100~~

Train empty checkpoint on factoid and list questions	{'factoid': {'avg': {'strict_accuracy': 0.2589354838709677, 'leniant_accuracy': 0.4514838709677421, 'mrr': 0.3171935483870968}, 'best': {'strict_accuracy': 0.312, 'leniant_accuracy': 0.562, 'mrr': 0.386}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}} {'factoid': {'avg': {'strict_accuracy': 0.2335483870967743, 'leniant_accuracy': 0.3470967741935485, 'mrr': 0.27670967741935487}, 'best': {'strict_accuracy': 0.32, 'leniant_accuracy': 0.44, 'mrr': 0.367}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}} {'factoid': {'avg': {'strict_accuracy': 0.15122580645161288, 'leniant_accuracy': 0.19567741935483882, 'mrr': 0.16490322580645161}, 'best': {'strict_accuracy': 0.179,

	'leniant_accuracy': 0.25, 'mrr': 0.198}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}} {'factoid': {'avg': {'strict_accuracy': 0.3253548387096773, 'leniant_accuracy': 0.529322580645161, 'mrr': 0.3895806451612903}, 'best': {'strict_accuracy': 0.412, 'leniant_accuracy': 0.588, 'mrr': 0.463}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}} {'factoid': {'avg': {'strict_accuracy': 0.33464516129032257, 'leniant_accuracy': 0.4526774193548387, 'mrr': 0.3776129032258063}, 'best': {'strict_accuracy': 0.406, 'leniant_accuracy': 0.5, 'mrr': 0.444}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}}
Train empty checkpoint on factoid questions sonly	{'factoid': {'avg': {'strict_accuracy': 0.24890322580645158, 'leniant_accuracy': 0.4043225806451612, 'mrr': 0.2970645161290322}, 'best': {'strict_accuracy': 0.312, 'leniant_accuracy': 0.5, 'mrr': 0.379}}} {'factoid': {'avg': {'strict_accuracy': 0.2400000000000001, 'leniant_accuracy': 0.34064516129032263, 'mrr': 0.2778709677419354}, 'best': {'strict_accuracy': 0.32, 'leniant_accuracy': 0.4, 'mrr': 0.347}}} {'factoid': {'avg': {'strict_accuracy': 0.14887096774193545, 'leniant_accuracy': 0.192225806451613, 'mrr': 0.16229032258064519}, 'best': {'strict_accuracy': 0.179, 'leniant_accuracy': 0.214, 'mrr': 0.19}}} {'factoid': {'avg': {'strict_accuracy': 0.27419354838709664, 'leniant_accuracy': 0.4886129032258063, 'mrr': 0.3470967741935483}, 'best': {'strict_accuracy': 0.382, 'leniant_accuracy': 0.559, 'mrr': 0.445}}} {'factoid': {'avg': {'strict_accuracy': 0.3235483870967742, 'leniant_accuracy': 0.4486129032258064, 'mrr': 0.3711935483870969}, 'best': {'strict_accuracy': 0.438, 'leniant_accuracy': 0.5, 'mrr': 0.459}}}
Train empty checkpoint on squad with learning rate 0.0003	{'factoid': {'avg': {'exact_match': 0.2819354838709677, 'f1': 0.4422580645161288 }, 'best': {'exact_match': 0.31, 'f1': 0.47}}}
Train empty	{'factoid': {'avg': {'exact_match': 0.2877419354838709, 'f1':

checkpoint on squad with learning rate 0.0001	0.4554838709677423}, 'best': {'exact_match': 0.3, 'f1': 0.46}}}
Train squad factoid checkpoint on bioasq factoid questions (99611)	{'factoid': {'avg': {'strict_accuracy': 0.26300000000000007, 'leniant_accuracy': 0.5049354838709678, 'mrr': 0.34158064516129033}, 'best': {'strict_accuracy': 0.312, 'leniant_accuracy': 0.562, 'mrr': 0.393}}} {'factoid': {'avg': {'strict_accuracy': 0.28516129032258075, 'leniant_accuracy': 0.3870967741935485, 'mrr': 0.3257419354838709}, 'best': {'strict_accuracy': 0.32, 'leniant_accuracy': 0.44, 'mrr': 0.361}}} {'factoid': {'avg': {'strict_accuracy': 0.20490322580645173, 'leniant_accuracy': 0.23729032258064522, 'mrr': 0.2149677419354839}, 'best': {'strict_accuracy': 0.25, 'leniant_accuracy': 0.286, 'mrr': 0.262}}} {'factoid': {'avg': {'strict_accuracy': 0.37564516129032255, 'leniant_accuracy': 0.5587419354838707, 'mrr': 0.4343225806451613}, 'best': {'strict_accuracy': 0.412, 'leniant_accuracy': 0.588, 'mrr': 0.464}}} {'factoid': {'avg': {'strict_accuracy': 0.3395806451612903, 'leniant_accuracy': 0.504, 'mrr': 0.39138709677419353}, 'best': {'strict_accuracy': 0.406, 'leniant_accuracy': 0.594, 'mrr': 0.465}}}
Train squad factoid checkpoint on bioasq factoid questions, this time printing our predictions (99625)	{'factoid': {'avg': {'strict_accuracy': 0.272, 'leniant_accuracy': 0.4779354838709678, 'mrr': 0.3426129032258065}, 'best': {'strict_accuracy': 0.312, 'leniant_accuracy': 0.531, 'mrr': 0.386}}} {'factoid': {'avg': {'strict_accuracy': 0.26451612903225824, 'leniant_accuracy': 0.37548387096774194, 'mrr': 0.3087741935483871}, 'best': {'strict_accuracy': 0.32, 'leniant_accuracy': 0.4, 'mrr': 0.353}}} {'factoid': {'avg': {'strict_accuracy': 0.2292258064516129, 'leniant_accuracy': 0.26399999999999985, 'mrr': 0.24341935483870966}, 'best': {'strict_accuracy': 0.25, 'leniant_accuracy': 0.286, 'mrr': 0.268}}} {'factoid': {'avg': {'strict_accuracy': 0.3766129032258064, 'leniant_accuracy': 0.5417096774193547, 'mrr': 0.43003225806451617}, 'best': {'strict_accuracy': 0.412, 'leniant_accuracy': 0.588, 'mrr': 0.468}}} {'factoid': {'avg': {'strict_accuracy': 0.3639032258064516, 'leniant_accuracy': 0.5160645161290324, 'mrr': 0.4101290322580645}, 'best': {'strict_accuracy': 0.406, 'leniant_accuracy': 0.594, 'mrr': 0.466}}}

Train squad factoid checkpoint on bioasq factoid and list questions, this time printing our predictions (99626)	Overall Metrics {'factoid': {'avg': {'strict_accuracy': 0.24699999999999994, 'leniant_accuracy': 0.5239677419354837, 'mrr': 0.34103225806451626}, 'best': {'strict_accuracy': 0.281, 'leniant_accuracy': 0.562, 'mrr': 0.381}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}} {'factoid': {'avg': {'strict_accuracy': 0.2967741935483872, 'leniant_accuracy': 0.4296774193548388, 'mrr': 0.34632258064516125}, 'best': {'strict_accuracy': 0.36, 'leniant_accuracy': 0.48, 'mrr': 0.401}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}} {'factoid': {'avg': {'strict_accuracy': 0.19925806451612912, 'leniant_accuracy': 0.24538709677419354, 'mrr': 0.21416129032258063}, 'best': {'strict_accuracy': 0.214, 'leniant_accuracy': 0.25, 'mrr': 0.226}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}} {'factoid': {'avg': {'strict_accuracy': 0.36338709677419345, 'leniant_accuracy': 0.5359999999999999, 'mrr': 0.42680645161290315}, 'best': {'strict_accuracy': 0.412, 'leniant_accuracy': 0.559, 'mrr': 0.462}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}} {'factoid': {'avg': {'strict_accuracy': 0.27599999999999997, 'leniant_accuracy': 0.5503225806451613, 'mrr': 0.36377419354838714}, 'best': {'strict_accuracy': 0.312, 'leniant_accuracy': 0.625, 'mrr': 0.409}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}}
Train empty checkpoint on bioasq factoid questions,	Overall Metrics {'factoid': {'avg': {'strict_accuracy': 0.303258064516129, 'leniant_accuracy': 0.45661290322580655, 'mrr': 0.3731612903225808}, 'best': {'strict_accuracy': 0.375,

collating predictions using our new technique. (99795) Learning rate = 1e-4	'leniant_accuracy': 0.531, 'mrr': 0.448}}} {'factoid': {'avg': {'strict_accuracy': 0.310967741935484, 'leniant_accuracy': 0.3870967741935486, 'mrr': 0.3389677419354838}, 'best': {'strict_accuracy': 0.44, 'leniant_accuracy': 0.44, 'mrr': 0.44}}} {'factoid': {'avg': {'strict_accuracy': 0.156741935483871, 'leniant_accuracy': 0.2304193548387097, 'mrr': 0.1866774193548387}, 'best': {'strict_accuracy': 0.25, 'leniant_accuracy': 0.25, 'mrr': 0.25}}} {'factoid': {'avg': {'strict_accuracy': 0.41935483870967744, 'leniant_accuracy': 0.5133548387096771, 'mrr': 0.4588064516129033}, 'best': {'strict_accuracy': 0.529, 'leniant_accuracy': 0.559, 'mrr': 0.544}}} {'factoid': {'avg': {'strict_accuracy': 0.3903870967741937, 'leniant_accuracy': 0.5008064516129032, 'mrr': 0.43890322580645136}, 'best': {'strict_accuracy': 0.469, 'leniant_accuracy': 0.531, 'mrr': 0.495}}}
Train empty checkpoint on bioasq factoid and list questions, collating predictions using our new technique. (99796) Learning rate = 1e-4	{'factoid': {'avg': {'strict_accuracy': 0.28619354838709676, 'leniant_accuracy': 0.45667741935483874, 'mrr': 0.35829032258064514}, 'best': {'strict_accuracy': 0.375, 'leniant_accuracy': 0.5, 'mrr': **0.438**}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}} {'factoid': {'avg': {'strict_accuracy': 0.2980645161290324, 'leniant_accuracy': 0.38709677419354843, 'mrr': 0.33112903225806456}, 'best': {'strict_accuracy': 0.4, 'leniant_accuracy': 0.48, 'mrr': **0.433**}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}} {'factoid': {'avg': {'strict_accuracy': 0.1338387096774193, 'leniant_accuracy': 0.21883870967741934, 'mrr': 0.16406451612903225}, 'best': {'strict_accuracy': 0.179, 'leniant_accuracy': 0.25, 'mrr': **0.208**}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}} {'factoid': {'avg': {'strict_accuracy': 0.4148064516129033, 'leniant_accuracy': 0.5181612903225804, 'mrr': 0.46187096774193553}, 'best': {'strict_accuracy': 0.529, 'leniant_accuracy': 0.559, 'mrr': **0.544**}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0,

	'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}} {'factoid': {'avg': {'strict_accuracy': 0.428483870967742, 'leniant_accuracy': 0.5068387096774191, 'mrr': 0.4604193548387096}, 'best': {'strict_accuracy': 0.5, 'leniant_accuracy': 0.594, 'mrr': **0.542**}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}}
Train squad checkpoint on bioasq factoid questions, collating predictions using our new technique. 99797 Learning rate = 1e-4	{'factoid': {'avg': {'strict_accuracy': 0.44174193548387086, 'leniant_accuracy': 0.5697096774193547, 'mrr': 0.49983870967741933}, 'best': {'strict_accuracy': 0.5, 'leniant_accuracy': 0.594, 'mrr': **0.539**}}} {'factoid': {'avg': {'strict_accuracy': 0.33677419354838717, 'leniant_accuracy': 0.46322580645161315, 'mrr': 0.3940645161290323}, 'best': {'strict_accuracy': 0.4, 'leniant_accuracy': 0.48, 'mrr': **0.44**}}} {'factoid': {'avg': {'strict_accuracy': 0.21422580645161296, 'leniant_accuracy': 0.27212903225806434, 'mrr': 0.23603225806451605}, 'best': {'strict_accuracy': 0.25, 'leniant_accuracy': 0.286, 'mrr': **0.268**}}} {'factoid': {'avg': {'strict_accuracy': 0.4402258064516132, 'leniant_accuracy': 0.546516129032258, 'mrr': 0.48654838709677434}, 'best': {'strict_accuracy': 0.529, 'leniant_accuracy': 0.559, 'mrr': **0.544**}}} {'factoid': {'avg': {'strict_accuracy': 0.44180645161290316, 'leniant_accuracy': 0.580774193548387, 'mrr': 0.5047741935483869}, 'best': {'strict_accuracy': 0.531, 'leniant_accuracy': 0.594, 'mrr': **0.562**}}}
Train squad checkpoint on bioasq factoid and list questions, collating predictions using our new technique. 99798 Learning rate = 1e-4	Overall Metrics {'factoid': {'avg': {'strict_accuracy': 0.41467741935483887, 'leniant_accuracy': 0.5419999999999999, 'mrr': 0.4667741935483872}, 'best': {'strict_accuracy': 0.438, 'leniant_accuracy': 0.562, 'mrr': **0.495**}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}} {'factoid': {'avg': {'strict_accuracy': 0.33806451612903227, 'leniant_accuracy': 0.45290322580645187, 'mrr': 0.39387096774193553}, 'best': {'strict_accuracy': 0.44, 'leniant_accuracy': 0.48, 'mrr': **0.46**}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}}

	{'factoid': {'avg': {'strict_accuracy': 0.1499677419354838, 'leniant_accuracy': 0.2686774193548385, 'mrr': 0.19387096774193543}, 'best': {'strict_accuracy': 0.179, 'leniant_accuracy': 0.286, 'mrr': **0.226**}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}} {'factoid': {'avg': {'strict_accuracy': 0.5198064516129033, 'leniant_accuracy': 0.5484838709677418, 'mrr': 0.5331935483870969}, 'best': {'strict_accuracy': 0.559, 'leniant_accuracy': 0.559, 'mrr': **0.559**}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}} {'factoid': {'avg': {'strict_accuracy': 0.39525806451612916, 'leniant_accuracy': 0.5534838709677418, 'mrr': 0.4659354838709679}, 'best': {'strict_accuracy': 0.469, 'leniant_accuracy': 0.594, 'mrr': **0.521**}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}}
Trying to increase the learning rate to 3e-4, experimenting with empty checkpoint on bioasq factoid 99801	{'factoid': {'avg': {'strict_accuracy': 0.40829032258064507, 'leniant_accuracy': 0.5444838709677418, 'mrr': 0.473709677419355}, 'best': {'strict_accuracy': 0.531, 'leniant_accuracy': 0.594, 'mrr': **0.562**}}} {'factoid': {'avg': {'strict_accuracy': 0.384516129032258, 'leniant_accuracy': 0.4348387096774196, 'mrr': 0.40345161290322573}, 'best': {'strict_accuracy': 0.48, 'leniant_accuracy': 0.48, 'mrr': **0.48**}}} {'factoid': {'avg': {'strict_accuracy': 0.13838709677419347, 'leniant_accuracy': 0.23151612903225804, 'mrr': 0.17641935483870966}, 'best': {'strict_accuracy': 0.214, 'leniant_accuracy': 0.25, 'mrr': **0.232**}}} {'factoid': {'avg': {'strict_accuracy': 0.4770645161290322, 'leniant_accuracy': 0.5351935483870965, 'mrr': 0.5046774193548389}, 'best': {'strict_accuracy': 0.529, 'leniant_accuracy': 0.559, 'mrr': **0.544**}}} {'factoid': {'avg': {'strict_accuracy': 0.4797741935483872, 'leniant_accuracy': 0.5399354838709676, 'mrr': 0.5067741935483874}, 'best': {'strict_accuracy': 0.562, 'leniant_accuracy': 0.562, 'mrr': **0.562**}}}
Trying to increase the learning rate to	{'factoid': {'avg': {'strict_accuracy': 0.3941612903225807, 'leniant_accuracy': 0.578741935483871, 'mrr': 0.474516129032258}, 'best': {'strict_accuracy': 0.438, 'leniant_accuracy': 0.594, 'mrr': 0.**508**}}}

3e-4, experimenting with squad checkpoint on bioasq factoid 99815	{'factoid': {'avg': {'strict_accuracy': 0.3380645161290323, 'leniant_accuracy': 0.4735483870967745, 'mrr': 0.40229032258064523}, 'best': {'strict_accuracy': 0.44, 'leniant_accuracy': 0.48, 'mrr': 0.**46**}}} {'factoid': {'avg': {'strict_accuracy': 0.21180645161290337, 'leniant_accuracy': 0.27554838709677404, 'mrr': 0.23558064516129026}, 'best': {'strict_accuracy': 0.25, 'leniant_accuracy': 0.286, 'mrr': 0.**262**}}} {'factoid': {'avg': {'strict_accuracy': 0.4706451612903226, 'leniant_accuracy': 0.551290322580645, 'mrr': 0.5083548387096775}, 'best': {'strict_accuracy': 0.559, 'leniant_accuracy': 0.559, 'mrr': 0.**559**}}} {'factoid': {'avg': {'strict_accuracy': 0.5229999999999999, 'leniant_accuracy': 0.5879032258064515, 'mrr': 0.5543870967741934}, 'best': {'strict_accuracy': 0.562, 'leniant_accuracy': 0.594, 'mrr': 0.**578**}}}
Trying to increase the learning rate to 3e-4, experimenting with squad checkpoint on bioasq factoid and list 99816	{'factoid': {'avg': {'strict_accuracy': 0.3860967741935485, 'leniant_accuracy': 0.5512258064516128, 'mrr': 0.460483870967742}, 'best': {'strict_accuracy': 0.469, 'leniant_accuracy': 0.562, 'mrr': 0.**516**}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}} {'factoid': {'avg': {'strict_accuracy': 0.2593548387096775, 'leniant_accuracy': 0.46709677419354867, 'mrr': 0.34993548387096773}, 'best': {'strict_accuracy': 0.32, 'leniant_accuracy': 0.48, 'mrr': 0.**393**}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}} {'factoid': {'avg': {'strict_accuracy': 0.1301612903225807, 'leniant_accuracy': 0.2604838709677418, 'mrr': 0.18103225806451603}, 'best': {'strict_accuracy': 0.214, 'leniant_accuracy': 0.286, 'mrr': 0.**239**}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}} {'factoid': {'avg': {'strict_accuracy': 0.42409677419354846, 'leniant_accuracy': 0.5561290322580643, 'mrr': 0.4818387096774193}, 'best': {'strict_accuracy': 0.559, 'leniant_accuracy': 0.559, 'mrr': 0.**559**}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}} {'factoid': {'avg': {'strict_accuracy': 0.42251612903225816, 'leniant_accuracy': 0.5726451612903224, 'mrr': 0.4893548387096775}, 'best': {'strict_accuracy': 0.531,

	'leniant_accuracy': 0.594, 'mrr': 0.**562**}}, 'list': {'avg': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}, 'best': {'mean_average_precision': 0.0, 'mean_average_recall': 0.0, 'mean_average_f1': 0.0}}}
Train empty checkpoint with enriched data 99857	Overall Metrics {'factoid': {'avg': {'strict_accuracy': 0.41445161290322574, 'leniant_accuracy': 0.5433870967741934, 'mrr': 0.47374193548387106}, 'best': {'strict_accuracy': 0.531, 'leniant_accuracy': 0.594, 'mrr': 0.**562**}}} {'factoid': {'avg': {'strict_accuracy': 0.3754838709677419, 'leniant_accuracy': 0.4516129032258067, 'mrr': 0.4090645161290324}, 'best': {'strict_accuracy': 0.48, 'leniant_accuracy': 0.48, 'mrr': 0.**48**}}} {'factoid': {'avg': {'strict_accuracy': 0.1696451612903226, 'leniant_accuracy': 0.2396451612903226, 'mrr': 0.19393548387096773}, 'best': {'strict_accuracy': 0.214, 'leniant_accuracy': 0.25, 'mrr': 0.**232**}}} {'factoid': {'avg': {'strict_accuracy': 0.47709677419354835, 'leniant_accuracy': 0.5361935483870965, 'mrr': 0.5046774193548388}, 'best': {'strict_accuracy': 0.559, 'leniant_accuracy': 0.559, 'mrr': 0.**559**}}} {'factoid': {'avg': {'strict_accuracy': 0.469741935483871, 'leniant_accuracy': 0.5565806451612902, 'mrr': 0.5070967741935484}, 'best': {'strict_accuracy': 0.562, 'leniant_accuracy': 0.594, 'mrr': 0.**578**}}}
Train empty checkpoint with further-enriched data 99857	Overall Metrics {'factoid': {'avg': {'strict_accuracy': 0.41448387096774186, 'leniant_accuracy': 0.5422258064516128, 'mrr': 0.4754838709677421}, 'best': {'strict_accuracy': 0.5, 'leniant_accuracy': 0.594, 'mrr': **0.547**}}} {'factoid': {'avg': {'strict_accuracy': 0.38064516129032255, 'leniant_accuracy': 0.4516129032258067, 'mrr': 0.41148387096774225}, 'best': {'strict_accuracy': 0.44, 'leniant_accuracy': 0.48, 'mrr': **0.46**}}} {'factoid': {'avg': {'strict_accuracy': 0.1659677419354839, 'leniant_accuracy': 0.23500000000000004, 'mrr': 0.19248387096774197}, 'best': {'strict_accuracy': 0.214, 'leniant_accuracy': 0.25, 'mrr': **0.232**}}} {'factoid': {'avg': {'strict_accuracy': 0.5036129032258064, 'leniant_accuracy': 0.543806451612903, 'mrr': 0.5220000000000001}, 'best': {'strict_accuracy': 0.559, 'leniant_accuracy': 0.559, 'mrr': **0.559**}}}

	{'factoid': {'avg': {'strict_accuracy': 0.4888064516129034, 'leniant_accuracy': 0.5339677419354837, 'mrr': 0.5092258064516131}, 'best': {'strict_accuracy': 0.562, 'leniant_accuracy': 0.562, 'mrr': **0.562**}}}
Train base model empty checkpoint on boolq 99424	
Train base model empty checkpoint on squad 99425	

Question Type: Yesno Script: 100249 Epochs: 30 Learning Rate: 1e-4 Dataset: BoolQ Chkpt: Empty	Overall Metrics {'yesno': {'avg': {'accuracy': 0.8920645161290326, 'precision': 0.8818387096774193, 'recall': 0.9809677419354834, 'f1_y': 0.9239999999999999, 'f1_n': 0.7887096774193553, 'f1_ma': 0.8564193548387098}, 'best': {'accuracy': 0.978, 'precision': 0.982, 'recall': 0.982, 'f1_y': 0.982, 'f1_n': 0.972, 'f1_ma': 0.977}}}
Question Type: Yesno Script: 100352 Epochs: 30 Learning Rate: 3e-4 Dataset: BoolQ Chkpt: Empty	{'yesno': {'avg': {'accuracy': 0.9056774193548387, 'precision': 0.9074838709677417, 'recall': 0.9715161290322577, 'f1_y': 0.9331935483870966, 'f1_n': 0.8120645161290324, 'f1_ma': 0.872709677419355}, 'best': {'accuracy': 0.968, 'precision': 0.966, 'recall': 0.982, 'f1_y': 0.974, 'f1_n': 0.957, 'f1_ma': 0.966}}}
Question Type: Yesno Script: Epochs: 30 Learning Rate: 3e-4 Dataset: BioASQ Chkpt: Empty	
Question Type: Yesno Script: 100406 Epochs: 30 Learning Rate: 3e-4 Dataset: BioASQ Chkpt: BoolQ lr 3e-4	
Question Type: Yesno Script: 100417 Epochs: 100 Learning Rate: 3e-4	{'yesno': {'avg': {'accuracy': 0.5275247524752464, 'precision': 0.6242475247524755, 'recall': 0.7543069306930699, 'f1_y': 0.6809702970297035, 'f1_n': 0.0497029702970297, 'f1_ma': 0.3655841584158419}, 'best': 0.54, 'epoch_of_best': 20}}

Dataset: BioASQ Chkpt: Empty	{'yesno': {'avg': {'accuracy': 0.725376237623763, 'precision': 0.7743861386138613, 'recall': 0.897425742574258, 'f1_y': 0.8301089108910897, 'f1_n': 0.2528712871287127, 'f1_ma': 0.5412673267326735}, 'best': 0.704, 'epoch_of_best': 44}} {'yesno': {'avg': {'accuracy': 0.6447623762376242, 'precision': 0.6511782178217828, 'recall': 0.8535148514851478, 'f1_y': 0.7353267326732676, 'f1_n': 0.4273960396039601, 'f1_ma': 0.581217821782179}, 'best': 0.689, 'epoch_of_best': 54}} {'yesno': {'avg': {'accuracy': 0.5713168316831683, 'precision': 0.5708217821782181, 'recall': 0.8259900990099008, 'f1_y': 0.672138613861387, 'f1_n': 0.3504059405940591, 'f1_ma': 0.5111683168316834}, 'best': 0.607, 'epoch_of_best': 50}} {'yesno': {'avg': {'accuracy': 0.7126336633663369, 'precision': 0.6872772277227722, 'recall': 0.9290198019801986, 'f1_y': 0.7860396039603957, 'f1_n': 0.5317425742574257, 'f1_ma': 0.6588118811881186}, 'best': 0.785, 'epoch_of_best': 20}}
Question Type: Yesno Script: 100419 Epochs: 100 Learning Rate: 3e-4 Dataset: BioASQ Chkpt: BoolQ lr 3e-4	
Question Type: Yesno Script: 100420 Epochs: 100 Batch Size: 32 Learning Rate: 3e-4 Dataset: BioASQ Chkpt: BoolQ lr 3e-4	
Question Type: Factoid List Epochs: 1 Dataset: BioASQ Chkpt: Empty	Overall Metrics {'factoid': {'avg': {'strict_accuracy': 0.094, 'leniant_accuracy': 0.281, 'mrr': 0.156}, 'best': {'strict_accuracy': 0.094, 'leniant_accuracy': 0.281, 'mrr': 0.156}, 'epoch_of_best': 0}, 'list': {'avg': {'mean_average_precision': 0.027, 'mean_average_recall': 0.35, 'mean_average_f1': 0.049}, 'best': {'mean_average_precision': 0.027, 'mean_average_recall': 0.35, 'mean_average_f1': 0.049}, 'epoch_of_best': 0}} {'factoid': {'avg': {'strict_accuracy': 0.04, 'leniant_accuracy': 0.12, 'mrr': 0.063}, 'best': {'strict_accuracy': 0.04, 'leniant_accuracy': 0.12, 'mrr': 0.063}, 'epoch_of_best': 0}, 'list': {'avg': {'mean_average_precision': 0.029, 'mean_average_recall': 0.357, 'mean_average_f1': 0.051},

	'best': {'mean_average_precision': 0.029, 'mean_average_recall': 0.357, 'mean_average_f1': 0.051}, 'epoch_of_best': 0}} {'factoid': {'avg': {'strict_accuracy': 0.143, 'leniant_accuracy': 0.143, 'mrr': 0.143}, 'best': {'strict_accuracy': 0.143, 'leniant_accuracy': 0.143, 'mrr': 0.143}, 'epoch_of_best': 0}, 'list': {'avg': {'mean_average_precision': 0.045, 'mean_average_recall': 0.417, 'mean_average_f1': 0.078}, 'best': {'mean_average_precision': 0.045, 'mean_average_recall': 0.417, 'mean_average_f1': 0.078}, 'epoch_of_best': 0}} {'factoid': {'avg': {'strict_accuracy': 0.059, 'leniant_accuracy': 0.324, 'mrr': 0.16}, 'best': {'strict_accuracy': 0.059, 'leniant_accuracy': 0.324, 'mrr': 0.16}, 'epoch_of_best': 0}, 'list': {'avg': {'mean_average_precision': 0.029, 'mean_average_recall': 0.588, 'mean_average_f1': 0.055}, 'best': {'mean_average_precision': 0.029, 'mean_average_recall': 0.588, 'mean_average_f1': 0.055}, 'epoch_of_best': 0}} {'factoid': {'avg': {'strict_accuracy': 0.156, 'leniant_accuracy': 0.25, 'mrr': 0.19}, 'best': {'strict_accuracy': 0.156, 'leniant_accuracy': 0.25, 'mrr': 0.19}, 'epoch_of_best': 0}, 'list': {'avg': {'mean_average_precision': 0.099, 'mean_average_recall': 0.5, 'mean_average_f1': 0.151}, 'best': {'mean_average_precision': 0.099, 'mean_average_recall': 0.5, 'mean_average_f1': 0.151}, 'epoch_of_best': 0}}
Question Type: Factoid List **101144** Epochs: 30 Dataset: BioASQ Chkpt: Empty	{'factoid': {'avg': {'strict_accuracy': 0.11080645161290323, 'leniant_accuracy': 0.27799999999999997, 'mrr': 0.1789032258064517}, 'best': {'strict_accuracy': 0.156, 'leniant_accuracy': 0.281, 'mrr': 0.219}, 'epoch_of_best': 5}, 'list': {'avg': {'mean_average_precision': 0.03277419354838711, 'mean_average_recall': 0.25806451612903225, 'mean_average_f1': 0.054709677419354834}, 'best': {'mean_average_precision': 0.06, 'mean_average_recall': 0.35, 'mean_average_f1': 0.095}, 'epoch_of_best': 8}} {'factoid': {'avg': {'strict_accuracy': 0.03225806451612903, 'leniant_accuracy': 0.1483870967741936, 'mrr': 0.08012903225806452}, 'best': {'strict_accuracy': 0.08, 'leniant_accuracy': 0.16, 'mrr': 0.12}, 'epoch_of_best': 19}, 'list': {'avg': {'mean_average_precision': 0.07048387096774195, 'mean_average_recall': 0.45864516129032257, 'mean_average_f1': 0.11235483870967741}, 'best': {'mean_average_precision': 0.095, 'mean_average_recall': 0.571, 'mean_average_f1': 0.15}, 'epoch_of_best': 14}} {'factoid': {'avg': {'strict_accuracy': 0.09780645161290329,

	'leniant_accuracy': 0.13722580645161284, 'mrr': 0.11490322580645161}, 'best': {'strict_accuracy': 0.143, 'leniant_accuracy': 0.179, 'mrr': 0.161}, 'epoch_of_best': 3}, 'list': {'avg': {'mean_average_precision': 0.05103225806451613, 'mean_average_recall': 0.39245161290322583, 'mean_average_f1': 0.0853225806451613}, 'best': {'mean_average_precision': 0.094, 'mean_average_recall': 0.5, 'mean_average_f1': 0.15}, 'epoch_of_best': 19}} {'factoid': {'avg': {'strict_accuracy': 0.11580645161290319, 'leniant_accuracy': 0.32183870967741923, 'mrr': 0.18516129032258072}, 'best': {'strict_accuracy': 0.147, 'leniant_accuracy': 0.382, 'mrr': 0.223}, 'epoch_of_best': 24}, 'list': {'avg': {'mean_average_precision': 0.04870967741935483, 'mean_average_recall': 0.5407096774193546, 'mean_average_f1': 0.08683870967741936}, 'best': {'mean_average_precision': 0.073, 'mean_average_recall': 0.588, 'mean_average_f1': 0.124}, 'epoch_of_best': 27}} {'factoid': {'avg': {'strict_accuracy': 0.14309677419354838, 'leniant_accuracy': 0.2960645161290322, 'mrr': 0.1984838709677419}, 'best': {'strict_accuracy': 0.188, 'leniant_accuracy': 0.312, 'mrr': 0.24}, 'epoch_of_best': 13}, 'list': {'avg': {'mean_average_precision': 0.10751612903225806, 'mean_average_recall': 0.3763548387096774, 'mean_average_f1': 0.15619354838709673}, 'best': {'mean_average_precision': 0.161, 'mean_average_recall': 0.417, 'mean_average_f1': 0.223}, 'epoch_of_best': 21}}
Question Type: Yesno **101178** Epochs: 30 Dataset: BioASQ Chkpt: Empty Eval on 9b1	{'yesno': {'avg': {'accuracy': 0.5029677419354841, 'precision': 0.4650967741935485, 'recall': 0.7633870967741934, 'f1_y': 0.5752580645161293, 'f1_n': 0.3790322580645162, 'f1_ma': 0.4770645161290324}, 'best': {'accuracy': 0.556, 'precision': 0.5, 'recall': 0.667, 'f1_y': 0.572, 'f1_n': 0.539, 'f1_ma': 0.555}, 'epoch_of_best': 4}}
Question Type: Yesno **101194** Epochs: 30 Dataset: BoolQ Chkpt: Empty High learning rate 3e-4 Eval on 9b1	{'yesno': {'avg': {'accuracy': 0.9056774193548387, 'precision': 0.9074838709677417, 'recall': 0.9715161290322577, 'f1_y': 0.9331935483870966, 'f1_n': 0.8120645161290324, 'f1_ma': 0.872709677419355}, 'best': {'accuracy': 0.968, 'precision': 0.966, 'recall': 0.982, 'f1_y': 0.974, 'f1_n': 0.957, 'f1_ma': 0.966}, 'epoch_of_best': 15}}
Question Type: Yesno **101196** Epochs: 100	Overall Metrics {'yesno': {'avg': {'accuracy': 0.9734950495049502, 'precision': 0.9707227722772274, 'recall': 0.9953861386138613, 'f1_y':

Dataset: BoolQ Chkpt: Empty High learning rate 3e-4 Eval on 9b1	0.9812079207920791, 'f1_n': 0.9490297029702969, 'f1_ma': 0.9649603960396038}, 'best': {'accuracy': 1.0, 'precision': 1.0, 'recall': 1.0, 'f1_y': 1.0, 'f1_n': 1.0, 'f1_ma': 1.0}, 'epoch_of_best': 25}}
Question Type: Yesno **101200** Epochs: 100 Dataset: BoolQ Chkpt: Empty Low learning rate 1e-4 Eval on 9b1	Overall Metrics {'yesno': {'avg': {'accuracy': 0.9485148514851495, 'precision': 0.9447524752475256, 'recall': 0.9877128712871286, 'f1_y': 0.9630495049504948, 'f1_n': 0.9045445544554459, 'f1_ma': 0.9336039603960394}, 'best': {'accuracy': 1.0, 'precision': 1.0, 'recall': 1.0, 'f1_y': 1.0, 'f1_n': 1.0, 'f1_ma': 1.0}, 'epoch_of_best': 52}}
Question Type: Yesno **101249** Epochs: 25 Dataset: BoolQ Chkpt: Empty Eval on 9b1	Match best of 101196
Question Type: Yesno **101258** Epochs: 100 Dataset: BioASQ Chkpt: **101249 checkpoint** Eval on 9b1	Overall Metrics {'yesno': {'avg': {'accuracy': 0.538217821782178, 'precision': 0.486207920792079, 'recall': 0.6781485148514849, 'f1_y': 0.5636039603960394, 'f1_n': 0.49816831683168306, 'f1_ma': 0.5308217821782176}, 'best': {'accuracy': 0.667, 'precision': 0.6, 'recall': 0.75, 'f1_y': 0.667, 'f1_n': 0.667, 'f1_ma': 0.667}, 'epoch_of_best': 67}}
Question Type: Factoid, List **101264** Epochs: 30 Dataset: BioASQ Chkpt: **Empty** Eval on 9b1	
Question Type: Factoid **101344** Epochs: 30 Dataset: Squad Chkpt: **Empty** **High lr**	
Question Type: Factoid **101329** Epochs: 100 Dataset: BioASQ	

Chkpt: **Empty**	
Question Type: List **101358** Epochs: 100 Dataset: BioASQ Chkpt: **Empty**	
Small model Squad checkpoint Epochs: 30 QT: factoid, list **101382**	THIS IS THE LAST SMALL CHECKPOINT
BASE Model Question Type: Yes No **101386** Epochs: 100 Dataset: Bioasq Chkpt: **Empty**	

Putting four models on:

- Small Bool Q: 101389 - small_yesno_18_64089_32_32
- Small Squad: 101391 - small_factoid_18_64089_5_365
- Base Bool Q: 101390 - base_yesno_1_104283_8_48
- Base Squad: 101392

Finetune checkpoint

- Small Yesno Bioasq (small_yesno_18_64089_32_32) : 101395 - **small_yesno_0_0_86_56 (downloaded)**
- Base Yesno Bioasq ("base_yesno_1_104283_8_48") : 101396 -
- Small Factoid,List Bioasq: 101405 - **small_factoid,list_0_0_24_0 (downloaded)**
- Base Factoid,List Bioasq: 101410 - base_factoid,list_1_104283_15_520 or base_factoid,list_1_104283_17_456

Putting four models on:

- Base yesno bioasq: 105690
- Small yesno bioasq: small_yesno_3_129918_29_103
- Small factoid list bioasq: 105621
- Base factoid list bioasq: 105622

FeatureUnion and Pipelines, give weights so how much the headline influences FeatureUnion.

Got BERT embedding in RNN

- Exploration
- Explanation
- Justification

Google scholar that dataset and see how it has been used.
Explore and discuss other people's ideas

Use the competition bodies as the validation set

80 20 for training and validation
Split training into training and validation

- Difference between test and validation? Test is what you test it on and validation tells you while training how the model is doing. We provide results in the test data. Test has to be the competition data.

Hyperparameters: - find justification e.g. "in line with this author we use this" and also play around with some different things (e.g. alpha for adam)

LSTMs are sequential and don't just have to take word vectors. Transformers give word embeddings.

Big sparse representations may be not good for CNNS. discuss why we're not using this in the paper. Hashing trick? Look into this?

How are TFIDF features used in LSTMS. TFIDF is bad for CNNs. What other papers have used tfidf in lstms - google for advantages and disadvantages. Find a complete comparison of different features and different model types.

Why is tfidf good for this dataset compared to others
Discuss how they work and advantages and disadvantages
Some of the advantages may change since we have a long document

Add a really small amount of related work

Plot the ROC curve

Ethical concerns:

- Companies may automatically try to gather reviews

15th March

Putting two models on:

- Small Bool Q: 102047 - small_yesno_18_64089_32_32
- Small Squad: 102048 - small_factoid_18_64089_17_441

add extra residual connections to the model to make sure info flow is there - no bottleneck preventing the gradients to flow. (random residual connections)

27th March

Evaluation:

Conclusion was clear. But it did not conclude many parts of the project, rather than it jumped to conclude the project used a bidirectional LSTM approach to solve the problem, which showed an incomplete picture of the project.

- Train electra small

Project Aims:

- Train a tokeniser on biomedical vocabulary
- Compare the results to BioBERT, a non-compute efficient extractive QA model.

Basic Objectives:

- Implement the ELECTRA model and pretrain to create bio
- Collect and analyse biomedical question-answering dataset
-

Intermediate Objectives:

- Explore the effect of fine-tuning on benchmark general-domain question-answering datasets before training on BioASQ.
- Submit to the BioASQ challenge to compare to current SOTA in 2021.

Advanced Objectives

- Build a user-friendly GUI

Instantiate weights and use normal tokeniser

-

Finetune Small on Yesno

PRETRAIN CHECKPOINT = recent FINETUNE CHECKPOINT = empty QUESTION TYPE = yesno DATASET = bioasq MODEL SIZE = small Learning rate=3e-4	Overall Metrics {'yesno': {'avg': {'accuracy': 0.625, 'precision': 0.625, 'recall': 1.0, 'f1_y': 0.7689999999999995, 'f1_n': 0.0, 'f1_ma': 0.38499999999999984}, 'best': {'accuracy': 0.625, 'precision': 0.625, 'recall': 1.0, 'f1_y':

	0.769, 'f1_n': 0, 'f1_ma': 0.385}, 'epoch_of_best': 0}}
PRETRAIN CHECKPOINT = recent FINETUNE CHECKPOINT = empty QUESTION TYPE = yesno DATASET = bioasq MODEL SIZE = small Learning rate=2e-4 107932	
107501 doubled weihts	Overall Metrics {'yesno': {'avg': {'accuracy': 0.6051612903225805, 'precision': 0.6644838709677419, 'recall': 0.842645161290323, 'f1_y': 0.7409354838709674, 'f1_n': 0.12538709677419352, 'f1_ma': 0.4330967741935486}, 'best': {'accuracy': 0.6, 'precision': 0.706, 'recall': 0.706, 'f1_y': 0.706, 'f1_n': 0.375, 'f1_ma': 0.54}, 'epoch_of_best': 9}} {'yesno': {'avg': {'accuracy': 0.696225806451613, 'precision': 0.7710645161290317, 'recall': 0.8496129032258062, 'f1_y': 0.8051290322580643, 'f1_n': 0.24419354838709678, 'f1_ma': 0.5243870967741935}, 'best': {'accuracy': 0.75, 'precision': 0.8, 'recall': 0.889, 'f1_y': 0.842, 'f1_n': 0.4, 'f1_ma': 0.621}, 'epoch_of_best': 11}} {'yesno': {'avg': {'accuracy': 0.6754516129032261, 'precision': 0.6695161290322583, 'recall': 0.9068064516129027, 'f1_y': 0.7663548387096772, 'f1_n': 0.4277741935483869, 'f1_ma': 0.596806451612903}, 'best': {'accuracy': 0.774, 'precision': 0.789, 'recall': 0.833, 'f1_y': 0.81, 'f1_n': 0.72, 'f1_ma': 0.765}, 'epoch_of_best': 9}} {'yesno': {'avg': {'accuracy': 0.6028064516129032, 'precision': 0.593483870967742, 'recall': 0.8709677419354837, 'f1_y': 0.7025483870967745, 'f1_n': 0.3640645161290322, 'f1_ma': 0.5331935483870968}, 'best': {'accuracy':

	0.692, 'precision': 0.65, 'recall': 0.929, 'f1_y': 0.765, 'f1_n': 0.556, 'f1_ma': 0.661}, 'epoch_of_best': 15}} {'yesno': {'avg': {'accuracy': 0.6233225806451612, 'precision': 0.6270967741935481, 'recall': 0.8454516129032253, 'f1_y': 0.7143225806451611, 'f1_n': 0.3953870967741936, 'f1_ma': 0.554774193548387}, 'best': {'accuracy': 0.676, 'precision': 0.7, 'recall': 0.737, 'f1_y': 0.718, 'f1_n': 0.621, 'f1_ma': 0.669}, 'epoch_of_best': 9}}
PRETRAIN CHECKPOINT = recent FINETUNE CHECKPOINT = empty QUESTION TYPE = yesno DATASET = boolq 107952	