Generalization Experiments + Research Plan

Overarching Question

Why do Transformer Language Models (TLMs) perform poorly on closed-domain datasets with "a focus" on Extractive QA?

*Note: Extractive QA is simply our application area of interest. It **does not** imply that TLMs are **particularly poor** on this task.*

Research Plan

Phase 1 – Exploration [Paper 1]

- **The *why?***
- Current Project - ***exploring multiple hypothesis*** to ***understand the phenomena***
- Building on the extended abstract submitted to NLDL/AAAI

Phase 2 – Solutions [2 papers]

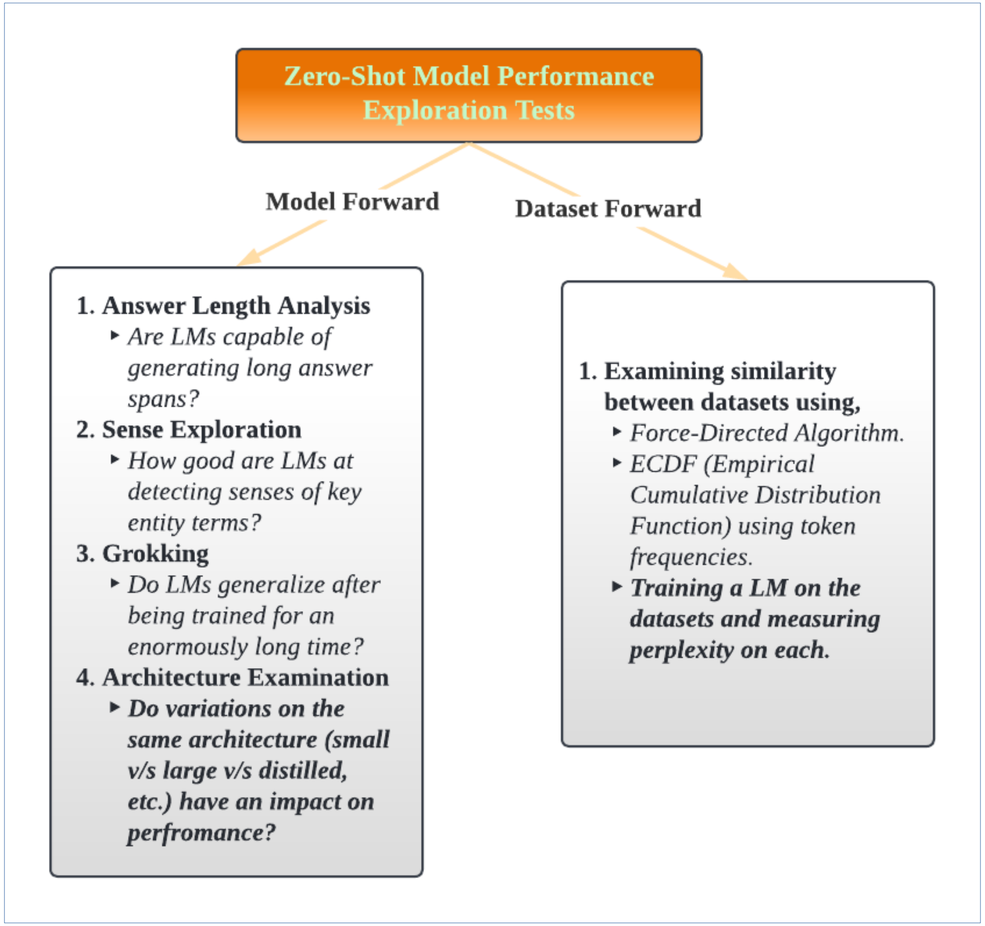
- **The *how?***
- How can we address the problems that were identified?

Adding **Knowledge Graph** Information to Transformers (our KGE paper)

Auxiliary Loss idea (Need to revisit that)

Phase 1 planned experiments

- Experiments for this phase are divided into 2 categories depending on the aspect that we're examining,
  - Model Forward [Issues in the architecture of the models]
  - Dataset Forward [Issues in the closed-domain datasets themselves]
  - Experiment 4 (model-forward) & 1c (dataset-forward) was suggested by Dr. Nakov



Datasets used – for Phase 1

- I discuss the datasets here since these are fairly new datasets (apart from DuoRC) and other papers on a related topic haven't utilized them.
- Open Domain – SQuAD
- Closed Domain –
  - DuoRC (Movies; Although the subject matter is believed to appear in the Wikipedia training corpus, the statistics of the dataset such as longer contexts and questions make this a candidate for Closed Domain)
  - TechQA (Technical domain – customer support on tech forums)
  - COVID-QA (COVID-19 related text)
  - CUAD (Legal domain)

Progress so far – Phase 1

- Model Forward
  - Predicted answer length scores obtained (Exp. 1)
  - Initial set of similarity scores (Exp. 2) obtained. However, I’m wondering whether this experiment needs reworking since I’m not too confident about the approach.
  - Grokking – TODO (Exp. 3)
  - Architecture Analysis – TODO (Exp. 4)
- Dataset Forward
  - All tests remaining.

Progress so far – Phase 2

- Regarding Auxiliary Loss Project,
  - Needs to be revisited.
- Regarding KGE Project,
  - We’ve come up with a new approach,
  - Motivation – Instead of taking an average of subword embeddings, we <u>want a single embedding</u> standing as the true projection i.e. Instead of avg. ([HIV] + [-] + [1] + [infection]), we’ll have, [HIV-1 infection]. This will go some way towards solving the out-of-vocabulary issue.
  - Run entity linker on contexts and obtain the top N frequent entities.
  - Initialize random embeddings for those entities and add them to the tokenizer.
  - Obtain contexts for those entities (scraping the internet) and train embeddings for the added tokens using the subword training scheme.
  - Train either the NN/Mikolov weight matrix, **[using the obtained context entity embeddings & their corresponding KGE’s]** and **use it to obtain target embeddings for the question entities** & append as previously done.