CS224N : Lecture 11 - Question Answering

What is question answering?

- **Goal of Question Answering**
    - To build systems that automatically answer questions posed by humans in a natural language
- **What information source does a system build on?**
    - A text passage, all Web documents, knowledge bases, tables, images
- **Question type**
    - Factoid vs non-factoid, open-domain vs closed-domain, simple vs compositional, ..
- **Answer type**
    - A short segment of text, a paragraph, a list, yes/no
- **Question answering in deep learning era**
    - Almost all the state-of-the-art question answering systems are built on top of endto-end training and pre-trained language models (e.g., BERT)
- **Beyond textual QA problems**
    - Mostly focus on how to answer questions based on unstructured text

Reading comprehension

- **Reading comprehension**
    - Comprehend a passage of text and answer questions about its content
    - $(P, Q) \longrightarrow A$

- **Why do we care about this problem?**
  - Useful for many practical applications
  - Reading comprehension is an important testbed for evaluating how well computer systems
    understand human language
  - Many other NLP tasks can be reduced to a reading comprehension problem
    - Information extraction
    - Semantic role labeling
- **Stanford question answering dataset (SQuAD)**
  - 100k annotated (passage, question, answer) triples
  - Passages are selected from English Wikipedia, usually 100~150 words
  - Questions are crowd-sourced
  - Each answer is a short segment of text (or span) in the passage
  - For development and testing sets, 3 gold answers are collected
- **BiDAF**
  - Use a concatenation of word embedding (GloVe) and character embedding (CNNs over character embeddings) for each word in context and query.
  - Use two bidirectional LSTMs separately to produce contextual embeddings for both context and query.
  - Context-to-query attention: For each context word, choose the most relevant words from the query words.
  - Query-to-context attention: choose the context words that are most relevant to one of query words.
  - Modeling layer: pass g to another two layers of bi-directional LSTMs
  - Output layer: two classifiers predicting the start and end positions
  - Model achieved 77.3 F1 on SQuAD v1.1
- **BERT**

- A deep bidirectional Transformer encoder pre-trained on large amounts of text (Wikipedia + BooksCorpus)
    - Pre-trained on two training objectives : MLM, NSP
    - BERT_base Has 12 layers and 110M parameters, BERT_large has 24 layers and 330M parameters
- **Comparisons between BiDAF and BERT models**
    - BERT model has many many more parameters (110M or 330M) and BiDAF has ~2.5M parameters
    - BiDAF is built on top of several bidirectional LSTMs while BERT is built on top of Transformers (no recurrence architecture and easier to parallelize).
    - BERT is pre-trained while BiDAF is only built on top of GloVe (and all the remaining parameters need to be learned from the supervision datasets)
    - BiDAF and other models aim to model the interactions between question and passage.
    - BERT uses self-attention between the concatenation of question and passage
- **Design better pre-training objectives**
    - Masking contiguous spans of words instead of 15% random words
    - Using the two end points of span to predict all the masked words in between = compressing the
      information of a span into its two endpoints

Open-domain

- **Open-domain Question Answering**
    - Don’t assume a given passage
    - Only have access to a large collection of documents (e.g., Wikipedia).
    - Don’t know where the answer is located
    - The goal is to return the answer for any
      open-domain questions

- **Retriever-reader framework**
  - Input: a large collection of documents
  - Output: an answer string A
  - Retriever = A standard TF-IDF information-retrieval sparse model (a fixed module)
  - Reader = a neural reading comprehension model that we just learned
    - Trained on SQuAD and other distantly-supervised QA datasets
- **Traning the Retriever**
  - Joint training
    - Each text passage can be encoded as a vector using BERT and the retriever score can be measured as the dot product between the question representation and passage representation
    - It is not easy to model as there are a huge number of passages
  - Dense passage retrieval (DPR) : train the retriever using question-answer pairs
- **Dense retrieval + generative models**
  - Recent work shows that it is beneficial to generate answers instead of to extract answers.