Theoretical Questions – Oren Sultan

May 23, 2023

1 Open Questions

1. Three QA datasets that use QA to annotate intrinsic concepts are:

    (a) **QASRL (Question-Answer Semantic Role Labeling):** comprehend the semantic roles of constituents in the sentence in the context of question-answering. In QASRL the model is first recognize the verb in the sentence, then recognize the arguments in the sentence (the answers) and generate a question for each argument which represent the argument (instead of discrete label like in SRL). As we learned, SRL is an intrinsic task, hence QASRL is a QA dataset to annotate intrinsic concept (the role labeling concept of constituents).

    (b) **Quoref (Questionable Reading Comprehension):** comprehend the coreferential reasoning capability. Coreferential reasoning refers to the ability to understand and resolve references to entities or objects mentioned in the context. It involves correctly identifying and connecting pronouns (such as "he," "she," "it") or other noun phrases to their corresponding antecedents (previous mentions) in the text. As we learned, coreference resolution is an intrinsic task, hence Quoref is a QA dataset to to annotate intrinsic concept (coreferential reasoning concept).

    (c) **QNLI (Question-answering NLI)**: assessing models' understanding of language semantics, logical reasoning, and inference capabilities. The model is presented with a pair of sentences: a premise (usually a context or passage) and a hypothesis (a statement or question). The task is to determine whether the hypothesis can be inferred from the premise (entailment), contradicted by the premise (contradiction), or is unrelated to the premise (neutral). As we learned, NLI is an intrinsic task, hence QNLI is a QA dataset to to annotate this intrinsic concept.

2. Two general approaches towards summarization are **Interactive summarization** and **Multi-document summarization**.

    (a) **Interactive summarization**

    i. **Task definition**: a summarization framework that combines automated techniques with user feedback to generate concise and informative abstractive summaries.

    ii. **Notable benchmarks**: I didn't found benchmarks but I found the paper of Shapira et al. "Evaluating Interactive Summarization: an Expansion-Based Framework" which developed end-to-end evaluation framework for expansion-based interactive summarization, which considers the accumulating information along an interactive session.

    iii. **inherent challenges presented by the task**:

    A. **Modeling User Intent**: Understanding and modeling user intent accurately is crucial in interactive summarization. Interpreting user queries, feedback, and preferences to capture their information needs and goals can be challenging, as user intent may be implicit or evolve over time.

    B. **Information Overload**: Interactive summarization often deals with large volumes of information, and users may have to navigate through extensive document collections or multiple iterations of summaries.

    (b) **Multi-document summarization**

    i. **task definition**: automatically generates a summary by selecting and combining important sentences from multiple source documents.

    ii. **notable benchmarks**:

    A. **Name of benchmark:** Multi-news **What domain do they annotate?** news article **How many samples?** ~45K training, ~5K validation, ~5K test

    B. **Name of benchmark:** DUC 2004 **What domain do they annotate?** news articles. **How many samples?** 500 test samples

    iii. **inherent challenges presented by the task**:

    A. **Information fusion**: Integrating information from multiple documents while ensuring coherence and avoiding redundancy.

B. **Redundant information**: multiple documents often contain overlapping information, leading to redundancy in the summary. The challenge is how to ensure that the summary provides a concise and non-repetitive representation.

C. **Coherence and Cohesion**: Maintaining coherence and cohesion across multiple source documents is crucial for producing a coherent summary. The challenge is to ensure smooth transitions between different document segments.

3. RNNs are not parallelizable since they have a sequential dependency – the hidden state at a given time step is calculated based on the input at that time step and the previous hidden state. On the other hand, in Transformers parallelization can be achieved in **some steps**. In **training**, in the **encoder**, it reads all sequence of tokens in parallel, and the self attention mechanism in the encoder can be computed in parallel for all input positions. The self-attention mechanism in the encoder allows each input position to attend to all other positions. This means that the encoding of each input position can be computed independently and in parallel, as there are no dependencies between different positions. In the **decoder,** teacher forcing allows for parallel computation since the model's predictions at each time step are not used as input for the subsequent time step, instead the ground truth. In **inference**, it is **not parallelizable**, since in the decoder it generates one token at a time, depend on the output generated so far, and the input to the decoder which is the output of the encoder (auto-regressive).

4. Problems:

   (a) I will use (A) Fine-tune ELECTRA-base. Option (B) of prompt tuning on 11B parameters can be problematic with only one GPU and no available money to purchase online services. With option (C) we will loss many of the 10K training samples, since in in-context learning the number of examples to feed is limited (in instructGPT the limitation is 4096 tokens for both the instruction text and the associated context). Also, using InstructGPT some of the budget will be spend on the license instead of utilizing it for computational resources.

   (b) By Fine-tuning BERT for this task: We take the two sentences, put a separator <sep> after each sentence, and run them through the model. The separator is inserted to the pre-training of BERT, it allows to deal with inputs of this kind (pair of sentences). In the end, as regular (also in one sentence) we can take the vector of the [CLS] token (a learned vector representation of a sentence, which is the first token in every sentence in BERT) and feed it to a classifier head, which will be trained to return the label (in our case whether the pair of sentences follow the same style or not).

   (c) Two reasons not to use ChatGPT:

      i. There is no scientific paper for ChatGPT (like for other open source models) so many details are missing on this model such as how exactly is was trained, on which datasets, etc. For example, we don't know if ChatGPT already trained on the dataset we examine or not.

      ii. It is not free, not always stable, and we should rely on a company (OpenAI) that can change the product, or close it in one day.

   (d) Two reasons to use ChatGPT:

      i. It is a powerful LLM, it shows very good performance (SOTA), can be a good baseline (still we don't have to compare to this baseline in our paper because of the reasons not to use that I mentioned)

      ii. We can use few-shot (sometimes one-shot is enough), we don't need to fine-tune and of course not to train from scratch on our samples, which is really fast and easy to use.

2 Coding Part

Link to GitHub Repo: `https://github.com/orensul/ANLPEx1`