Commonsense Reasoning and Inference: Commonsense-Based Pre-Training

Pei Zhou
Department of Computer Science
University of Southern California
peiz@usc.edu

Abstract
Commonsense reasoning has been regarded as one of the hardest problems for artificial intelligence systems to solve, but recently there has been some breakthrough using pure deep representation models. Namely, pre-trained language representation models, such as BERT, perform quite well on Natural Language Understanding (NLU) tasks, including those that require commonsense reasoning. However, there is still a lot improvement space to both reach human-level performance and increase the interpretability of these models. Some recent efforts have been devoted to augmenting the pre-trained Language Models (LMs) with commonsense knowledge. This lecture will briefly introduce how LMs are used for commonsense tasks and then will discuss models that aim to augment them with commonsense-based pre-training.

1 Using Language Representations for Commonsense Reasoning

1.1 Motivation
In recent years, pre-trained LMs have made some of the most exciting breakthroughs in the field of Natural Language Processing (NLP) and numerous new models are designed to tackle different tasks. Commonsense reasoning, as one of the most important NLU tasks, has also been used to show LMs’ potency. Language models are pre-trained on large amounts of unsupervised corpora, which may contain certain degree of human commonsense knowledge. The two papers that will be introduced below are both conceptually very simple and show that with or even without fine-tuning, LMs can outperform previous state-of-the-art models by a large margin on a commonsense coreference task called Winograd Schema Challenge (WSC) [Levesque et al., 2012].

1.2 Task Description
WSC proposes a coreference resolution task that requires commonsense reasoning. The datasets provides a sentence with a pronoun, and asks the machine to find the right candidate for the pronouns from two options. Example sentence pairs are shown in Figure 1.

1.3 Methods

1.3.1 A Simple Method for Commonsense Reasoning
Overview They use language models (LMs), to score multiple choice questions posed by the challenge and similar datasets. More concretely, in the example question: “The trophy doesn’t fit in the suitcase because it is too big. What is too big? Candidates: 1. the trophy, 2. the suitcase.”, they will first substitute the pronoun (“it”) with the candidates (“the trophy” and “the suitcase”), and then
Figure 1: Some examples from Winograd Schema Challenge.

| I(a) | The city councilmen refused the demonstrators a permit because they feared violence. |
| I(b) | The city councilmen refused the demonstrators a permit because they advocated violence. |
| II(a) | James asked Robert for a favor, but he refused. |
| II(b) | James asked Robert for a favor, but he was refused. |
| III(a) | Keith fired Blaine but he did not regret. |
| III(b) | Keith fired Blaine although he is diligent. |
| IV(a) | Emma did not pass the ball to Janie, although she was open. |
| IV(b) | Emma did not pass the ball to Janie, although she should have. |
| V(a) | Medvedev will cede the presidency to Putin because he is more popular. |
| V(b) | Medvedev will cede the presidency to Putin because he is less popular. |

Table 1: Sample twin sentences. The target pronoun in each sentence is italicized, and its antecedent is boldfaced.

Figure 2: Overview of their method and analysis. They consider the test “The trophy doesn’t fit in the suitcase because it is too big.” Their method first substitutes two candidate references trophy and suitcase into the pronoun position. They then use an LM to score the resulting two substitutions. By looking at probability ratio at every word position, they are able to detect “big” as the main contributor to trophy being the chosen answer. When “big” is switched to “small”, the answer changes to suitcase. This switching behaviour is an important feature characterizing the Winograd Schema Challenge.

use LMs to compute the probability of the two resulting sentences (“The trophy doesn’t fit in the suitcase because the trophy is too big.” and “The trophy doesn’t fit in the suitcase because the suitcase is too big.”). The substitution that results in a more probable sentence will be the correct answer. Further analysis shows that their system successfully discovers the special word in the sentence like “big” to make its decisions in many cases, indicating a good grasp of commonsense knowledge.

Model Details They first substitute the pronoun in the original sentence with each of the candidate choices. The problem of coreference resolution then reduces to identifying which substitution results in a more probable sentence. By reframing the problem this way, language modeling becomes a natural solution by its definition. Namely, LMs are trained on text corpora, which encodes human knowledge in the form of natural language. During inference, LMs are able to assign probability to any given text based on what they have learned from training data. An overview of our method is shown in Figure 2.

They consider two different ways of scoring the substitution as shown in Figure 3.

1.3.2 A Surprisingly Robust Trick for WSC

Overview Kocijan et al. [2019] extend the previous work by fine-tuning BERT [Devlin et al. [2018]] on Winograd-like datasets and get even better results. One of the training objectives of BERT is
masked word prediction and they utilize this fact by masking the pronoun in WSC and ask BERT to predict the right word. To get more data for fine-tuning, they generate Winograd-like datasets from Wikipedia. Results show that they can improve upon previous SOTA methods by around 8%.

**Method**

Given a training sentence $s$, the pronoun to be resolved is masked out from the sentence, and the LM is used to predict the correct candidate in the place of the masked pronoun. Let $c_1$ and $c_2$ be the two candidates. BERT for Masked Token Prediction is used to find $\mathbb{P}(c_1|s)$ and $\mathbb{P}(c_2|s)$. If a candidate consists of several tokens, the corresponding number of [MASK] tokens is used in the masked sentence. Then, $\log \mathbb{P}(c_1|s)$ is computed as the average of log-probabilities of each composing token. If $c_1$ is correct, and $c_2$ is not, the loss is:

$$L = -\log \mathbb{P}(c_1|s) + \alpha \cdot \max(0, \log \mathbb{P}(c_2|s) - \log \mathbb{P}(c_1|s) + \beta),$$

where $\alpha$ and $\beta$ are parameters.

**MaskedWiki Dataset**

To get more data for fine-tuning, they automatically generate a large scale collection of sentences similar to WSC. More specifically, their procedure searches a large text corpus for sentences that contain (at least) two occurrences of the same noun. They mask the second occurrence of this noun with the [MASK] token. Several possible replacements for the masked token are given, for each noun in the sentence different from the replaced noun.

1.4 Some critics

**Interpretability**

Pure LM-based approaches for commonsense reasoning tend to be lacking of reasonable explanations since they can be mainly considered as black boxes.

**Robustness**

People have shown that WSC dataset contains bias of different kinds, from gender bias [Zhao et al. 2018], [Rudinger et al. 2018] to statistical bias [Trichelair et al. 2019]. And LMs will exploit them as a shortcut to do commonsense reasoning.

2 Augmenting Language Models by Incorporating Commonsense

2.1 Motivation

Neural language representation models such as Bidirectional Encoder Representations from Transformers (BERT) [Devlin et al. 2018] can well capture rich language information from unlabelled text, and can be fine-tuned to benefit many NLP applications. However, the existing pre-trained language representation models rarely consider explicitly incorporating commonsense knowledge or other knowledge. Specifically, the training objectives of LMs: masked word prediction and next sentence prediction do not incorporate knowledge reasoning. Thus, most recent work has studied ways to teach LMs with commonsense knowledge in the pre-training step as an additional objective.

2.2 Methods

2.2.1 Align, Mask and Select

**Overview**

They Ye et al. [2019] propose a pre-training approach for incorporating commonsense knowledge into language representation models. They construct a commonsenselated multi-choice question answering dataset for pretraining a neural language representation model. The dataset is created automatically by our proposed “align, mask, and select” (AMS) method. They also investigate different pretraining tasks.
Constructing Pre-Training Datasets  They first filter the triples in ConceptNet as follows: (1) Filter triples in which one of the concepts is not English words. (2) Filter triples with the general relations “RelatedTo” and “IsA”, which hold a large proportion in ConceptNet. (3) Filter triples in which one of the concepts has more than four words or the edit distance between the two concepts is less than four. After filtering, they obtain 606,564 triples. Each training sample is generated by three steps: align, mask and select, which is denoted the AMS method. Each sample in the dataset consists of a question and five candidate answers, which has the same form as the CommonsenseQA dataset. An example of constructing one training sample by masking concept2 is shown in Figure 4. Firstly, they align each triple (concept1, relation, concept2) in the filtered triple set to the English Wikipedia dataset to extract the sentences containing the two concepts. Secondly, they mask the concept1 or concept2 in one sentence with a special token [QW] and treat this sentence as a question, where QW is a replacement word of the question words “what”, “where”, etc. And the masked concept1 or concept2 is the correct answer for this question. Thirdly, for generating the distractors, Sun et al. [2019] formed distractors by randomly picking words or phrases in ConceptNet. In their work, in order to generate more confusing distractors than the random selection approach, they select distractors sharing the same other unmasked concept, i.e., concept2 or concept1, and the same relation with the correct answer. That is to say, they search (∗, relation, concept2) or (concept1, relation, ∗) in ConceptNet to select the distractors, where ∗ is a wildcard character that can match any word or phrase. For each question, we reserve four distractors and one correct answer. If there are less than four matched distractors, we discard this question instead of complementing it with random selection.
Pre-Training BERT

They investigate a multi-choice QA task for pre-training the English BERT base and BERT large models released by Google. The objective function is defined as follows:

$$L = - \log p(c_1 | s),$$

$$p(c_1 | s) = \frac{\exp(w^T c_1)}{\sum_{k=1}^{N} \exp(w^T c_k)},$$

where $c_1$ is the correct answer, $w$ the parameters in the softmax layer, $N$ the total number of candidates, and $c_1$ the vector representation of the special token [CLS].

2.2.2 Teaching Pretrained Models with Commonsense Reasoning

Overview

The key idea of their Li et al. [2019] method is to generate multiple-choice questions from different subgraphs in KB, and then they use the generated data to further refine the pretrained models. The overall idea of the data generation process is shown in Figure 5, which consists of (i) generating different logical forms from a sampled subgraph in KB, (ii) generating multiple-choice questions in natural language form.

Generating Logical Forms

They first sample a subgraph from KB that is in the following form:

$$(A \xrightarrow{R_1} B \xrightarrow{R_2} C),$$

where A, B, and C are three different entities in the KB, and $R_1$ and $R_2$ represent two different relations in the KB. For each of the above subgraph, they will construct a multiple-choice question regarding the entity B in the following manner. First, introduce the following two sets:

$$R_1 = \{ X \in \Omega : A \xrightarrow{R_1} X \}, \quad R_2 = \{ X \in \Omega : X \xrightarrow{R_2} C \},$$

where \( \Omega \) denotes the entire entity set. Note that the set $R_1$ represents the set of all (tail) entities that have relation $R_1$ with A, and $R_2$ represents the set of all (head) entities that have relation $R_2$ with entity C. Note from Figure 5 that the entire space could be partitioned into four subsets, denoted as: $S_1 = R_1 \cap R_2$, $S_2 = R_1 \cap R_2^c$, $S_3 = R_1^c \cap R_2$, $S_4 = R_1^c \cap R_2^c$. Each subset represents a certain logical relation. For example, the subset $S_2 = R_1 \cap R_2^c$ means all the entities that have relation $R_1$ with A and have relation $R_2$ with C. Using these four subsets, they could compose questions that ask about all different logical relations from the subgraph in the equation above. To see this, note that we could compose a set by either choosing or not choosing each subset $S_i$, which leads to a total of $2^4 = 16$ subsets. Among them, two trivial cases are excluded: the all-chosen case (full set) and the all-not-chosen set (empty set). Therefore, there are a total of 14 different logical relations about the equation above that they could ask.

Generating multiple-choice questions

They can generate natural language questions that ask about this particular logical relation. They achieve this by using text templates. Specifically, they first create two different types of mapping, namely, affirmative mapping and negative mapping. The affirmative mapping is used to generate sentences with affirmative questions, while the negative mapping is used for generating negative ones. Consider the specific example of a logical form in Figure 5, where the correct answer for the missing entity is people. In the above logical form, the relation CapableOf will be mapped into “is capable of” using affirmative mapping. On the other hand, when there is a negation before the relation CapableOf, it will be mapped into “is not capable of” using a negative mapping. These obtained strings from relations will be put together with the head entities and the tail entities to generate sentences as natural as possible by using a set of simple
heuristic rules. For example, the above logical relation will be mapped into the following natural language sentence: “which of the following is an antonym of alone and meanwhile is capable of sing in church?”

**Generating candidate answers** They will examine three different sampling strategies. The first approach is to random sample from the all the other entities. The second one is the nearest sampling. The third sampling method is uniform sampling: it firstly chooses wrong subset uniformly from $S_1, ..., S_4$ and then samples an entity from the selected subset.

**Teaching the pre-trained model with commonsense** To teach the pretrained models with commonsense reasoning, they further train the pretrained models on the generated multiple-choice questions to predict the correct answer, which becomes a multi-class classification problem. Afterwards, the model is finetuned on different downstream tasks. They name this step as refinement to distinguish it from the pretraining and the finetuning stages.

**References**


