BECON: BERT with Evidence from CONceptNet for Commonsense Question Answering

Motivation

CommonsenseQA dataset is created by crowdsourcing workers based on knowledge graphs on ConceptNet. Solving the tasks requries the model to have commonsense knowledge. Current LM-pretrained model such as BERT achieves SOTA performance on CQA dataset, which implies that language models trained on very large corpus may learn some commonsense implicitly. With the availability of the large knowledge graph such as ConceptNet, which contains explicit commonsense knowledge, we would like to know if we can use the *explicit* form of commonsense knowledge as a complementary to BERT which learns implicit commonsense knowledge.

Dataset

The statistics about the dataset is shown in the table.

Train	Dev	Test
9741	1221	1140

A Sample of the dataset is shown below.


```
{
    "answerKey": "B",
    "id": "70701f5d1d62e58d5c74e2e303bb4065",
    "question": {
        "choices": [
            {
                "label": "A",
                "text": "bunk"
            },
            {
                "label": "B",
                "text": "reading"
            },
            {
                "label": "C",
                "text": "think"
            },
            {
                "label": "D",
```

```
                "text": "fall asleep"
            },
            {
                "label": "E",
                "text": "meditate"
            }
        ],
        "stem": "What is someone doing if he or she is sitting quietly and his or her eyes are moving?"
    }
}
```


Evidence Finder

Each question has 5 candidate answers. According to our analysis, candidate answers are usually one or two words long. We can use ConceptNet API `http://api.conceptnet.io/c/en/` to find all the related information related with a word or phrase in the knowledge graph.

Example:


```
{
    "text": "meeting",
    "evidence": [
        "*Something you find at a meeting is notepad",
        "*Something you find at a meeting is an agenda",
        "*Something you find at a meeting is a group of people",
        "*Something you find at a meeting is discussion",
        "a stranger is for meeting",
        "appointment is related to meeting",
        "interview is related to meeting",
        "group meeting is a synonym of meeting",
        "rendezvous is a type of meeting",
        "*Something you find at a meeting is papers"
    ],
}
```


We expect that such evidences may be helpful to answer the question. The problem is, the evidence is too noisy. How to extract useful information? We would like to keep the evidence which is relevant to the question, and discard others. Assume that at most 1 evidence sentence is helpful (which means 0 or 1). We can first rank the evidences and then use the top-ranked evidence (or not).

Evidence Ranker

The evidence ranker ranks the evidences according to the relevant scores with the question. We consider some of the very simple rankers:

- **random**: no ranking. Just random shuffle.

- **jaccard**: Jaccard Index is a metrics which consider the words "intersection over union" between question and evidence sentences.
- **w2v**: the cosine distance between the average of pretrained word2vec embeddings of question and evidence sentences.
- **BERT**: use pretrained BERT model along with its Next Sentence Prediction head to determine the relevancy of two sentences.

To have a sense of how the rankers work, we use the ranker to rank all the evidences of the 5 candidate answers. The candidate answer with the top ranked evidence is chosen as the predicted answer.

A simple model without training: select the choice with the highest **evidence** score.

The result on ***train*** and ***dev*** is shown in the table below.

Ranker	train	dev	train_SANITY	dev_SANITY
random	21.14	19.82	21.07	19.57
jaccard	23.12	22.44	44.43	41.28
w2v	26.05	23.91	48.73	47.01
bert-base	34.95	34.73	82.89	81.90
bert-large	36.50	36.86	84.41	82.88

For comparison, below is the results on ***test*** from original paper on test split.

Models	test	test_SANITY
VECSIM+NUMBERBATCH	29.1	54.0
LM1B-REP	26.1	39.6
LM1B-CONCAT	25.3	37.4
VECSIM+GLOVE	22.3	26.8
BERT-LARGE	55.9	92.3
GPT	45.5	87.2
ESIM+ELMO	34.1	76.9
ESIM+GLOVE	32.8	79.1
QABILINEAR+GLOVE	31.5	74.8
ESIM+NUMBERBATCH	30.1	74.6
QABILINEAR+NUMBERBATCH	28.8	73.3
QACOMPARE+GLOVE	25.7	69.2
QACOMPARE+NUMBERBATCH	20.4	60.6
BIDAF++	32.0	71.0
HUMAN	88.9	-

There is no dev result in the original paper, but if we assume the dev and test result are close, we can see that the BERT-large NSP model without training is only inferior than BERT-large and GPT which use the CQA dataset to train.

This encourages us to think about another simple model without training: select the choice with the highest NSP score with the question. Below are the results.

NextSentencePrediction Pretrained BERT Model

Model	train	dev	train-SANITY	dev-SANITY
BERT-base NSP	35.36	39.39	71.28	71.99
BERT-large NSP	38.41	40.38	73.54	73.14

Still reletively high compared with the trained model, especially on "SANITY" variant of the dataset. It may indicates that the contribution from BERT model mainly comes from the "pretrain" phase.

Models

Literature & Baseline

Leaderboard

Models	test	test-SANITY
KagNet	**58.9**	
CoS-E	58.2	
BECON(ours)	**57.9+**	
SGN-lite	57.1	
BERT-large(Tel-Aviv U)	56.7	
BERT-large	55.9	92.3
BERT-base(UCL)	53.0	
GPT	45.5	87.2
ESIM+ELMo	34.1	76.9
ESIM+glove	32.8	79.1

Reproduce baseline

Models	dev
BERT-base	57.6
BERT-large	63.4

Our Model: BECON

For each answer candidate, rank evidences, and use top evidence.

[CLS] + Question + [SEP] + Answer + [SEP] + Evidence + [SEP]

Pretrain Models	ranker	dev
BERT-base	BERT-base	56.2
BERT-base	BERT-large	57.6
BERT-large	BERT-base	61.9
BERT-large	BERT-large	62.2

The comparision between BERT-base/large rankers show that BERT-large ranker is better. The experiments later all use BERT-large ranker.

Compared with our baseline, the result is a bit lower. This means if we add evidence for each answer candidate, the noise may still overwhelms the useful information.

Solution: Encode `BERT(Question + Answer)` as well as `BERT(Question + Answer + Evidence)`, and then use max/mean/concatenation as representation of the candidate answer.

Pretrain Models	pooling	dev
BERT-large	max	63.6
BERT-large	mean	64.0
BERT-large	concat (no pooling)	**64.4**

The concatenation without pooling outperforms the BERT-large baseline on dev by 1.0%.

We also try another way to incorprate the evidence: rank evidences among all candidate answers, and use the top-ranked evidence. We expect that in this way, since we use only 1 evidence for this sample, the noise will be lower.

`BERT(Question + Evidence + Answer)`

Question + Evidence Models	dev
BERT-base	58.3
BERT-large	62.8

It works on BERT-base (+0.7%), but not on BERT-large (-0.6%).

Discussion: BERT-Sep

`BERT(question) + BERT(answer) (+ BERT(evidence) ) << BERT(question + answer + evidence)`

Which means attention is not straight-forward to use in BERT models.

Model	dev
question + choice	22.4
question + choice + top-evidence	22.7

Summary

We use conceptnet to search for evidence, use BERT to rank them, and use BERT as the base model to train the model with evidence. To alleviate the noise introduced by the evidence, we use BERT to encode both w/ w/o evidence, and let model learn to choose which one contributes more. This model outperforms BERT-large baseline by 0.9% on dev and +1.2% on test, which proves the effectiveness of our method. For comparison, salesforce research use human

generated explaination to enhance the question, which only outperforms our model by 0.3%.

Another interesting phenomenon is that BERT NSP without any training on CQA dataset has comparable performance with ESIM + ELMO/glove models on CQA dataset.

NOTE: our submitted model is lower than our best model on dev by 0.4%. However we cannot resubmit until two weeks later. So if the increase is likewise on test set, we may have comparable results with COS-E by Salesforce Research.

Reference and Links

- Leaderboard (our result is #3)
- Dataset paper
- COS-E by Salesforce
- ConceptNet