The Death of Feature Engineering ? — BERT with Linguistic Features on SQuAD 2.0

Yue Zhang (yzhang16), Jiawei Li (jiaweili)

Background

- Machine reading comprehension is an essential NLP task, it is useful both for application and as a measure of how good NLP models understand given text.
- **Input:** A pair of context and query
- **Prediction:** The corresponding answer to query.

Dataset

- **Dataset:** Stanford Question Answering Dataset **2.0** [4], extending SQuAD 1.0 by adding questions that have no answer in the given text.
- **train set** (129,941 exampales): All taken from the o□cial SQuAD 2.0 training set.
- **dev set** (6078 examples): Roughly half of the o□cial dev set, randomly selected.
- **test set** (5921 examples): The remaining examples from the o□cial dev set, plus hand-labeled examples.

Motivation

- State-of-the-art model (BERT) performs well, close to human level, but still have some NLU errors.
- We propose incorporating linguistic features to help.

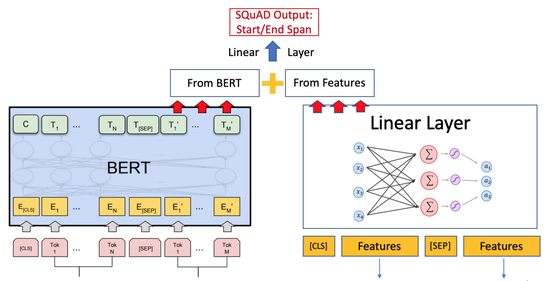
The Yuan dynasty is considered both a successor to the Mongol Empire and an imperial Chinese dynasty. It was the khanate ruled by the successors of Möngke Khan after the division of the Mongol Empire. In official Chinese histories, the Yuan dynasty bore the Mandate of Heaven, following the Song dynasty and preceding the Ming dynasty. The dynasty was established by Kublai Khan, yet he placed his grandfather Genghis Khan on the imperial records as the official founder of the dynasty as Taizu.[b] In the Proclamation of the Dynastic Name (《建國號詔》), Kublai announced the name of the new dynasty as Great Yuan and claimed the succession of former Chinese dynasties from the Three Sovereigns and Five Emperors to the Tang dynasty.

What dynasty came before the Yuan?
Ground Truth Answers: Song dynasty | Mongol Empire | the Song dynasty
Prediction: Ming dynasty

What dynasty came after the Yuan?
Ground Truth Answers: Ming dynasty | Ming dynasty | the Ming dynasty
Prediction: Song dynasty

Model Architecture

- **Input**: a pair of sentences: context c and question q
- **Output**: answer for the question (a span on the context sentences): start and end token index x_{start}, x_{end}
- **Metrics**: Exact Match (EM) score and F1 score
- **Model**: *BERT* [3, 1] and *Linguistic Feature model*



BERT Model

- Input features from tokenizer
  `(input_idx, mask, segment) = tokenizer(c,q)`
- Sequential Embedding Features from BERT
  `seq_out = BERT(input_idx, mask, segment)`

Linguistic Feature Model

- Linguistic features from Linguistic model
  `ling_out = linguistic_model(c,q)`

Output Layer

- Concatenate the output from both model
  `output_logits = output_layer(seq_out, ling_out)`

Linguistic Features

4 linguistic features are extracted for each token with NLP package SpaCy [2], the first 3 features encoded as integers.

- NER: Name entity label
- POS: The part-of-speech tag
- DEP: Syntactic dependency, token relationship.
- STOP: Is the token part of a stop list, 0/1 vector.

Performances

Table 1: Experiments Results for Single Model

Model Name	Dev Set	Test Set
Single Model	**EM/F1**	**EM/F1**
BiDAF	49.07/50.29	-/-
BERT (base)	71.59/74.72	-/-
Our model (base)	73.76/76.86	-/-
BERT (large)	78.51/81.34	-/-
Our model (large)	78.17/81.20	**76.55/79.97**

- **Improvement:** Our model improved EM score and F1 score by 2.17 and 2.14 compared with BERT in (both use bert base model)
- **Best Results** Our single best model reaches EM score 76.55 and F1 score 79.97

Results analysis

Table 2: Model Predictions

Context Part: Yuan dynasty		
Question	What non-Chinese empire did the Yuan dynasty succeed?	Which tribes did Genghis Khan fight against?
Reference	Mongol Empire	No Answer
BERT-feature	Mongol Empire	Mongol and Turkic
BERT	No Answer	Mongol and Turkic

- With the addition of extracted linguistic features, we find that the new model understands context with more complex linguistic structures better and is able to find the correct answer when BERT itself predicts "No Answer" wrongly.
- Our model still fails to make the correct prediction when the reference result is 'No answer'
- The major bottleneck of the current model comes from how we determine whether the answer exists for a certain question

Confusion Matrix

Table 3: Confusion Matrix for the Existence of the Answer

Confusion Matrix	**Answer**	**No Answer**
Answer	1456	1454
No Answer	1556	1612

- Very high **false positive** and **false negative** are observed here.
- our current model is unable to e□ectively make determine the **existence of the answer**, even though we have reached a fair high metric on EM/F1.

Conclusion and Future Work

- Adding features help increase performance of BERT base, no significant improvement for BERT large.
- We conclude "**Feature engineering is not dying**", especially when computational resources are not very cheap today.
- In future, we are interested in modifying the architecture and the loss function to get better results on the Answer / No Answer classification problem, multitasking learning is also a good candidate for making improvement on that.

References

[1] `https://github.com/huggingface/pytorch-pretrained-BERT`.

[2] `https://spacy.io/usage/linguistic-features`.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] P. Rajpurkar, R. Jia, and P. Liang. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*, 2018.