Machine Reading Comprehension on Clinical Case Reports (CliCR)

Jaehyuk Choi
Computing, Informatics & Decision Systems Engineering
Arizona State University
jchoi154@asu.edu

I. INTRODUCTION

Machine comprehension of question-answering task heavily depends on datasets. Challenges come from understanding the linguistic difficulties and developing new algorithms. Unlike open-domain datasets based on news, Wikipedia, etc., machine comprehension skills should be improved more for specific domain datasets. This project is to motivate researchers for machine comprehension of healthcare and medical documents. Given the data set of clinical case reports for the reading comprehension task, the team was motivated to improve existing NLP question-answering model by using our own techniques based on Bert. The team have used CliCR data set[1] which contains queries, answers and supporting passages from 12,000 BMJ Case Reports 2005-2016, the largest online repository of such documents for this project.

The main task was to answer the given queries by using a machine trained from understanding the supporting passages and extracting keywords. These multiple gap-filling queries are associated with a supporting passage which provides medical information for training a machine. Fig1 shows an example of supporting passage, a gap-filling query, and its answer.

The passage in Fig1 contains detailed clinical information such as diseases, unusual presentation of common conditions, and that of new treatments. The red sentence in the passage is key pieces of information to answer the given query. Paraphrasing such sentences with masking medical entities are called *Learning points*. The *Learning points* were rebuilt in order to reduce the potential errors and inconsistencies coming from automated dataset creation[1].

Answers of entity to each query was expanded with respective synonyms from UMLS Metathesauras (a biomedical thesaurus), including its CUI (Concept Unique Identifier); instead of *relapse*, it could be *recurrence* (C0035020).
Moreover, it is worth noting that the authors also provided answer entity candidates for each query in the dataset. However, due to our formulation, we were able to create a model that solves for the given query without utilizing that extra information, which is a more practical real life scenario for any Reading Comprehension Problem.

Passage: ... A gradual improvement in clinical and laboratory status was achieved within 20 days of antituberculous treatment . The patient was then subjected to a thoracic CT scan that also showed significant radiological improvement . Thereafter, tapering of corticosteroids was initiated with no clinical relapse. The patient was discharged after being treated for a total of 30 days and continued receiving antituberculous therapy with no reported problems for a total of 6 months under the supervision of his hometown physicians ...

Query: If steroids are used, great caution should be exercised on their gradual tapering to avoid _____.

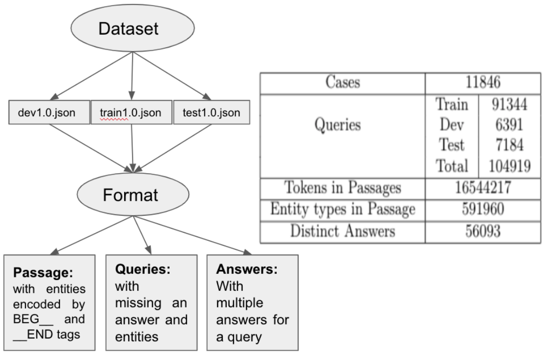
Answer: relapse (sem_type=problem, cui= C0035020)

Fig. 1. An Example of supporting passage, a gap-filling query, and its answer from the BMJ Case Reports dataset

II. DATASET

The dataset consists of passage, multiple queries for each supporting passage and set of answers as respective synonyms for each query; 11,846 cases, 104,919 queries, and 56,093 distinct answers. 59% instances were explicitly found in the relevant supporting passages. Fig2 shows number of cases, queries, tokens in passages, entity types in passages, and distinct answers.

The answers were categorized into 3 types; Problem, Treatment and Test. From 59% cases, 67% were problems like abdominal pain, acute myocardial infarction, etc; 22% were treatments like surgical intervention, vitamin D supplement, etc; 11% were tests like MRI, histopathological exam, etc.


Dataset
dev1.0.json
train1.0.json
test1.0.json
Format
Passage: with entities encoded by BEG__ and __END tags
Queries: with missing an answer and entities
Answers: With multiple answers for a query

Cases		11846
Queries	Train	91344
	Dev	6391
	Test	7184
	Total	104919
Tokens in Passages		16544217
Entity types in Passage		591960
Distinct Answers		56093

Fig. 2. Analyzed Dataset

The given query and supporting passage in Fig1 is simplified without tagging, but its actual format is shown in Fig3.

> Query1: "If BEG___steroids___END are used, great caution should be exercised on their gradual tapering to avoid @placeholder."

Fig. 3. Example of gap-filling query with medical entities

All medical entities in passages and queries were tagged with BEG___ and ___END tags to be recognized. Each query has @placeholder as the gap that machine should predict learning from a supporting passage.

Therefore, pre-processing was required; pairing (p, q) where p is a supporting passage excluding the Learning points; q is a query built from a learning point; and A is the set of ground-truth entities answering q and NER (Named Entity Recognition). The answer entities were extracted by using NER so that our solution only attempted to highlight the answer(s) relevant to a given query within a supporting passage, skipping those answers outside of the supporting passage.

III. Solutions

The original models mainly based on word2vec embeddings. Such a traditional word embedding builds a global vocabulary even though meaning of words can vary depending on contexts. Then, similar representations are learnt for the words appeared more frequently close each other in the documents. This narrow learning prevent understanding a passage[2].

However, a contextual embedding like BERT builds flexible vocabulary based on contexts, learning sequence-level semantics. Thus, such techniques learn various representations for polysemous words. Moreover, BERT finds semantic relationships within the input sequences in both directions, using multi-head attention and positional embedding.

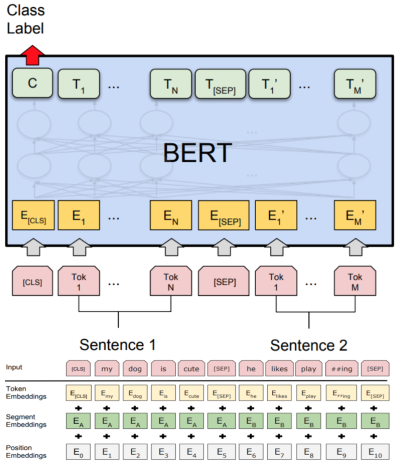


Fig. 4. BERT Sentence Pair Classification

A. Phase1

Our Phase1 model required pre-processing; NER tagging and pairing (p, q) each case report (passage) with associated queries and entities of ground-truth answers. Using BIO-schema (i.e. B-ans for the beginning of answer entities, I-ans for rest of answer entities, and O for others), NER tagging helps this question-answering task transformed into a classification task. Paring (p, q) has around 56,000 instances by considering answers explicitly stated in a supporting passage out of 93,000. The answer set includes respective synonyms to each query from UMLS. Due to the BERT size limit (max input sequence 512 tokens), we modify pairing; splitting a supporting passage p into 200 tokens and padding correspondent query q to the end of every part of p. To fine-tune our model, we used BertForTokenClassification class, wraping a pre-trained BERT model and adding Fully Connected Classification layer via HuggingFace Tranformers. Thus, the Fully Connected Classification layer works in the last state from BERT for every token and outputs the probability of every token tagged BIO-schema.

The problem comes from splitting p; Each query needs contextual meanings around the key sentence (which contains key medical information for a query) for its answer, but if those key sentences are not seen or partially broken into pieces while training, the model cannot understand the supporting passage enough to answer the given query. Let a full supporting passage be split into many piece such as P=p_1, p_2, p_3, ... In the former case that model cannot see key sentences, the model misses contextual connection between partial passage p_1 and the following partial passage p_2. In the latter case that partially broken into pieces, the key sentences may contain no longer important medical information.

B. Phase2

After Every pair (p, q) was encoded, collate function was applied to pass batches to the training loop. Then, every batch has a size of 1 instance; each instance works as one batch. This Dynamic Sequencing technique processes the collated pair (p, q) which were dynamically sent to BERT so that BERT generates the embedding for collated pair. These embeddings were concatenated as 1 dimension.

This concatenated embedding passed 1 dimensional Convolutional layer whose window size is 21 for classification of token's tag. The Convolutional layer seeing over partial pairs, helps BERT to update its weights for understanding over all contextual meaning through backpropagation of loss via BERT's embedding.

However, Dynamic Sequencing has limitation of size; 1 paragraph at a time. Thus, Gradient Accumulation helps the model to learn contextual meanings of (p, q) pairs in a bigger window because it backpropagates the loss after accumulation of gradients in a fixed steps. These gradients in a fixed steps encompasses learning contextual meanings for a bigger window of partial passages. This fixed number of steps reduces the randomness in the gradient direction per update so that it accelerates the entire training process. The number of fixed steps per update was selected as 16 for tuning. Thus, most effective batch size was 16 instances.

BERT is pre-trained on the concatenation of two huge corpora; BookCorpus and English Wikipediawas. These two corpora are useful for plain texts, not domain specific one. We used Bio-BERT, the pre-trained BERT model based on medical domain datasets. Compared to BERT, Bio-BERT has different weights. Bio-BERT was applied to the Phase1 model and the Phase2 model to see how it improves the base performance.

IV. RESULTS

The Table I shows evaluated models' performances including the original models (OG), human's performance (Humans), and our models (Phase1 and Phase2).

Based on the best performance of original models, we evaluate our solutions. The key score is F1-score. The training set, validation set, and testing set were randomly distributed. The testing set was intact and not spoiled; it remained only for testing purpose in order for reliable evaluation of testing performance.

The original models used Neural readers such as Stanford Attentive Reader (SA) and Gated Attention Reader (GA)[1]. The GA-Anonym performs the best among the original models as 0.33 F1-score. From the analysis of Phase1 and Phase2 models, we expected Phase2 model would outperform others. As TableI shows, the Phase2 BERT model performed 0.44 F1-score nearly close to human novice level. However, Phase2 with Bio-BERT diminished its performance. This is probably comes from internal compatibility error because before training, we converted Bio-BERT weights which trained through Tensorflow to a Pytorch readable format. Also, the Phase1 model with BERT and that with Bio-BERT performed the same as 0.36 F1-score. These counterintuitive outcomes were not studied further due to lacking time and computational cost of training to see the different results with modification.

	Model	F1-Score
OG model	GA-Anonyms	0.33
	GA-NoEnt	0.34
Human	Novice	0.45
	Expert	0.54
Phase1	BERT	0.36
	Bio	0.36
Phase2	BERT	0.44
	Bio	0.34

TABLE I

Evaluated Model Performance base on F1-Score

V. CONTRIBUTIONS

My contribution to the project was

- NER tagging of input sequences annotated with BIO-schema and pairing (p, q) where 200 tokens in p, followed by BERT special token [SEP], followed by a correspondent query q due to the size limitation of BERT.
- Padding all the tokenized sequences to the same maximum length and creating attention masks to explicitly differentiate real tokens from [PAD] tokens.
- Debugging and testing our algorithms on Agave Cluster.
- Dynamic Sequencing and Gradient Accumulation, and CNN layer;

VI. NEW SKILLS/TECHNIQUES/KNOWLEDGE

The project was for not only having practical experience but also improving theoretical analysis of NLP. Using open-source tools, I realized that various NLP techniques were melted in them.

Moreover, it requires devouring tremendous papers and tools' specifications/instructions especially Pytorch and Transformer (including BERT as part of it). Also, original models were based on Neural readers such as Stanford Attentive Reader and Gated Attention Reader based on RNNs. It was enormous project, which forced me learning modern NLP techniques and improved prioritizing tasks in limited time.

Team Member: Jaehyuk Choi, Abhik Dey, Eswar Gundabolu, Rishabh Jain, and Ahmed Kobtan.

References

[1] Simon Suster, Walter Daelemans. CliCR: A Dataset of Clinical Case Reports for Machine Reading Comprehension. NAACL-HLT 2018

[2] JacobDevlin, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.", ACL, 2019.

[3] Jinhyuk Lee et al., "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," Bioinformatics, vol. 36, no. 4, pp. 1234-1240, February 2020.