Team : Quarantine.ai
Pablo Marino
Zhizhen Wang
Vishay Vanjani

Literature Review

OVERVIEW

We chose papers in the Information Extraction domain for this task. These papers cover a range of topics, from popular neural network architectures, to ones that describe novel Information Extraction systems.

Papers

1. Attention is all you need[3]
2. Roberta[15]
3. Distilling knowledge in a neural network[7]
4. Generating question-answer hierarchies[10]
5. A simple but effective method to incorporate Multi-turn context with BERT for Conversational Machine Comprehension[13]
6. The third PASCAL recognizing Textual Entailment Challenge[6]
7. ReQA: An evaluation for end to end answer retrieval models[1]
8. A BERT baseline for natural questions[4]
9. TransferTransfo: A Transfer Learning Approach for Neural Network Based Conversational Agents[11]

Task Definition

Our team is interested in the question answering task, also known as “Machine comprehension”. The selected papers tackle all the subjects we think we need to understand in order to work on that problem: Transformers and Attention, BERT, Roberta, QA benchmarks like ReQA, implementation of QA systems using BERT(current state of the art) and data augmentation for QA.

Summaries

Attention is all you need

This seminal paper introduces a new neural network architecture called "transformer" that uses attention mechanism and feed forward networks, to solve the machine translation task. The performance of this new network architecture is much better than recurrent and convolutional architectures for machine translation (later Bert proved that it can get SOTA results on GLUE, SQUAD and other benchmarks as well). The biggest strength of this architecture is that it's easy to parallelize and hence can be executed efficiently on the GPU.

The transformer architecture relies entirely on the self-attention mechanism to compute representations of its input and output. It's two major components are the encoder transformer and the decoder transformer (similar to rnn seq2seq architecture). The encoder takes in the source language sentence and spits out a dense representation of the sentence. The decoder takes in that representation and spits out the translated sentence in German (the paper used WMT 2014 English-German and WMT 2014 English-French datasets).

The transformer uses scaled dot product attention: Softmax($QK^T/\sqrt{d_k}$)*V to calculate how much a word will be weighted when computing the next representation of the current word (Q is the linearly transformed representation of the word whose next representation is being calculated, K,V are the linearly transformed representation of the word in the context of the word whose next representation is being calculated, d_k is dimension of the embedding). Instead of performing a single attention function the authors decided to apply attention "h" times for the same input sentence. This was done to account for the fact that a word can mean different things to different neighbors and multiple attention heads allow the network to learn these different relations. The authors employed 8 attention heads. The paper "A primer in Bertalogy"[2] discovered that not all these attention heads convey non-trivial linguistic information for BERT. This model achieves SOTA BLEU score of 28.4 beating the previous best by 2.0 BLEU.

The transformer architecture introduced in this paper was the seed for the "ImageNet" moment in NLP. A majority of the best performing models that deliver SOTA results on NLP/NLU benchmarks (some of which surpass humans) use this architecture.

RoBerta

This paper is a replication study of BERT Pretraining, it also proposes modifications in BERT to establish new SOTA for GLUE(for 4/9 tasks), SQUAD and RACE benchmarks. The original BERT paper[9] used the encoder Transformer to create a new language model and was pre-trained on

the book corpus and English Wikipedia. The unsupervised pretraining phase for BERT had two objectives 1) Masked language modeling 2) Next sentence prediction.

The authors of Roberta prove empirically that the second training objective aka next sentence prediction is not essential and in some cases actually degrades the model performance for some downstream tasks. The MLM objective is the cross-entropy loss on predicting the masked tokens.

In the paper, the authors propose a more effective masking strategy aka “dynamic masking” where they generate a new mask pattern for each sequence. This ensures that no sequence is fed to the transformer with the same mask even in consecutive epochs (in BERT each training sequence was seen with the same mask 4 times for 40 epochs).

The other finding in this paper was that larger mini-batch sizes (during pre-training) perform better with BERT. They tried pre-training BERT with increased batch sizes of 2k,8k and found that batch size of 2k sequences with 125K steps gave the best results. The model was pre-trained with dynamic masking on full sentences without NSP loss (next sentence prediction) with larger mini-batch sizes on 1024 Nvidia V100 GPU’s for approximately 1 day.

Distilling Knowledge in a neural network

The paper describes a technique for compressing the knowledge of large neural network models (e.g. ensemble models) to make it easy to deploy (under resource constraints) and reduce the e2e inference latencies in real world/commercial scenarios (e.g. mobile phones, search engines).

They call the technique of transferring knowledge of the teacher model (heavyweight model with lots of parameters & long training time) to a student model (small model with less parameters) “Distillation/Knowledge Distillation”. Distillation is possible because when we train models, the optimization goal is to reduce the loss on the training set and not to generalize well on new data. For e.g. the ensemble (teacher) model can generalize well because it is the average of a large ensemble of models even when individual models (that are part of the ensemble) do not. A small model trained on the output of the teacher model (instead of the original labels/ground truths) will do much better on the test/new data (generalize well) than if it was trained in the normal way. The output of the final layer(softmax layer) of the teacher model aka the class probabilities (on which the student is trained) are referred to as soft targets.

The training objective for the student is 1) cross entropy with the soft targets 2) cross entropy with the ground truths/hard targets. For the first training objective, the paper suggests using high value of Temperature(in softmax calculation) for both the student and the teacher models. For the second objective they suggest using T=1. The below equation shows how temperatures is used in class probability(softmax layer) calculation (q_i is the class probability of class i, z is the corresponding logit (o/p of pre-softmax layer), T is temperature):

$$q_i = exp(z_i/T) / \sum_j exp(z_j/T)$$

The original paper used ensemble model as teacher and proved that student model can generalize well with only a small loss in accuracy. More recent works[12,14] have applied this approach to transformer models with great success. These new methods[12,14] use two phase knowledge distillation process, phase one involves pre-training and phase two involves fine-tuning (similar to the teacher).

A Simple but Effective Method to Incorporate Multi-turn Context with BERT for Conversational Machine Comprehension

BERT, as one of the most advanced contextual word representation frameworks, can be fine-tuned to adapt multiple tasks, and it is performing well on a single turn machine comprehension task. This paper explores the method that can apply BERT on conversational machine comprehension (CMC).

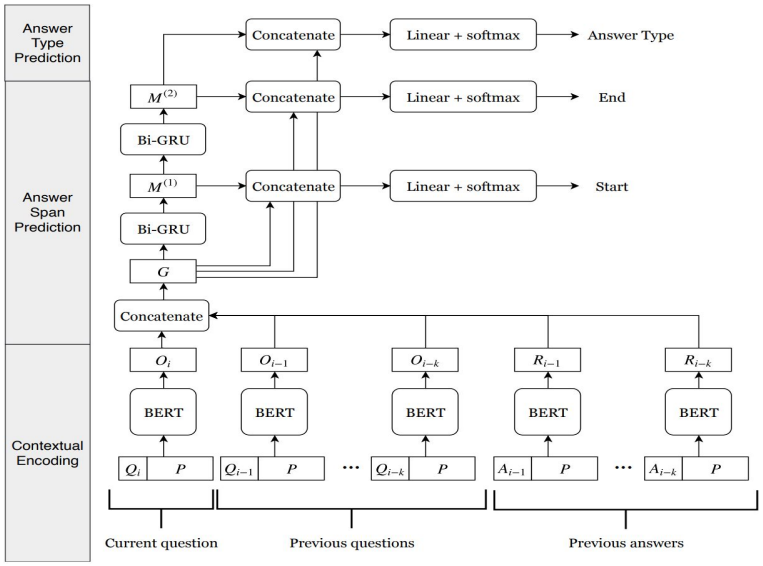
Answering multiple questions in a dialog is a practical task for any virtual agent system (Siri or Google Assistant). It requires the model to understand beyond the current round of Q&A and carry the answer history in the multi-turn dialogue to answer sequential questions.

However, BERT has a limit on the number and the length of input sequences because it can only accept two sequences of 512 tokens. Whereas in machine comprehension, the general method of fine-tuning BERT is to concatenate the questions and answers as two sequences in the pre-train process, then ask the fine-tuned BERT to predict on the relation of the unlabeled sequences in the downstream task.

Reformulating CMC into a single-turn MC task didn't yield a good result, because BERT cannot capture the interactions between questions-answering turns.

This paper proposed a new way to design the training to fine-tune BERT for CMC. At a higher level, the original method involved two steps. The first is to construct a set of training data of questions and paragraphs that can contain multiple relationships, including questions to answer, sequential questions, and answer history. Then in the second step, for a given question, ask BERT to extract the answer span from the corresponding paragraph.

The model structure is illustrated in the diagram below.



Generating Question-Answer Hierarchies

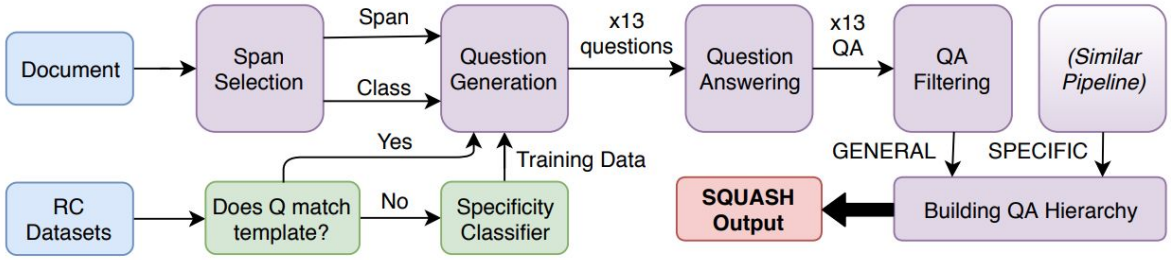
Structure of a training set and re-formulating the metrics to evaluate the tasks-oriented model result are critical to any NLP related experiments. This paper presents a new task called SQUASH (Specificity-controlled Question-Answer Hierarchies), which would automatically derive a hierarchy of question-answer pairs from any given documents.

This SQUASH task would potentially help with the Q&A tasks that did not have sufficient sequential question-answer pairs to build and extend their training set. Or it could help with information retrieval tasks and even dialog and knowledge acquisition.

As an overview, there are a few critical elements in building the new system:

1. **Hierarchy Tree** - The Q&A hierarchy tree was defined by a higher-level general question at the top and specific factoids at the bottom.
2. **Training Set** - The training set of this system was built by combining and restructuring several reading comprehension datasets(including SQuAD, QuAC, and CoQA), and then using the Taxonomy from Lehnert (1978) to label the relations between the questions.
3. **Pipeline for SQUASH** - The pipeline has five main functions, (1) answer span selection, (2) question generation conditioned on answer spans and specificity labels, (3) generate answers for questions, (4) filtering out bad QA pairs, and (5) structuring the remaining pairs into a GENERAL-to-SPECIFIC hierarchy.
4. **Evaluation** - Crowdsourced evaluation was chosen to assess the quality, relevance, and correctness of the QA Pair.

A more visual presentation of the pipeline was listed below,



The original deep learning model that generated the questions was an encoder and decoder mechanism with a two-layer LSTM encoder and a single layer LSTM decoder with soft attention. Later the team incorporated the newest language model like BERT and GPT2-small to further boost the performance of SQUASH.

Overall the SQUASH system presents a viable way to generate sequential Q&A pairs in unlabel documents to help on various NLP tasks in the downstream.

TransferTransfo: A Transfer Learning Approach for Neural Network Based Conversational Agents

TransferTransfo is a new approach proposed by the HuggingFace(https://huggingface.co/) team to tackle the problem of building data-driven dialogue systems(aka. ChatBot).

At a high level, this new approach mirrors the scheme of transfer learning, which incorporates using a powerful general-purpose model and fine-tunes the last few layers to accommodate new domains and new tasks.

Compared with the existing state-of-the-art end-to-end conversational models, TransferTransfo had a considerable improvement in terms of performance consistency, long term conversational memory, and produced a meaningful and targeted response.

At its core, TransferTransfo was a multi-layer Transformer encoder network, based on the previous research by Radford(from OpenAI team). It used a concatenated positional embeddings, dialog state embeddings, and word embeddings to process the input sentence, and the maximum tokens are up to 512. In the fine-tuning step, the team is using the PERSONA-CHAT dataset (a crowd-sourced dialogue dataset). And to achieve the multi-task objective, the team

stack several unsupervised prediction tasks including (i) a next-utterance classification, and (ii) a language modeling.

TransferTransfo outperforms the existing systems by a significant margin on the public validation dataset obtaining 51% absolute improvement in perplexity (PPL), 35% absolute improvement in Hits@1 and 13% improvement in F1.

The Third PASCAL Recognizing Textual Entailment Challenge

The paper talks about how the ability to recognize entailment between texts, ie: the ability of given 2 sentences T and H tell if H is a generalization of T, can be used as an engine for common NLP applications, before this field was developed each of these applications would develop their own specific algorithms/tasks to deal with semantic variability in language even though the problem was the same for all of them, NLI provides a common engine for all of them:

QA system: Identify texts that entail a hypothesized answer, given a question Q we convert it to an affirmation H and given a text T. An NLI classifier can be used to detect if the question is a generalization of the given text.
Eg:
Q: Was Lincoln born in the USA?
H: Lincoln was born in the USA
T: Future president Abraham Lincoln was born in Hodgenville, Kentucky, USA on February 12, 1809.

Information retrieval queries: The query will be used as hypothesis H and should be entailed by the retrieved documents T.

Multi-document summarization: Redundant sentences are omitted from the summary because they can be entailed from other sentences in the summary.

Machine translation evaluation: A correct translation should be semantically equivalent to the gold standard translation, thus both translations should entail each other.

This paper introduces a dataset for NLI binary classification problem, the dataset contains text T , hypothesis H and a boolean label indicating whether H entails T, the distribution of the labels is balanced(50% true and 50% false). The dataset is divided across 7 applications:
Information retrieval, comparable documents, reading comprehension, question answering, information extraction, Machine translation, paraphrase acquisition.

The dataset looks very high quality, but it's too small for today's data hungry neural networks: 800 examples in the training set and 567 in the development set.

ReQA: an evaluation for end to end answer retrieval models

This paper introduces a benchmark for evaluating end-to-end answer retrieval models. The task assesses how well models are able to retrieve relevant sentence-level answers to queries from a large corpus.

For evaluating the models they use SQUAD and NQ datasets, they recommend using different datasets(and not including data from wikipedia since these models are built from it) for training to guarantee the model can adapt to data drawn from new contexts.

Current successful QA systems usually follow a 2-step approach to answer a question: 1- Retrieve relevant articles or blocks. 2- Scan the returned text to identify the answer using a reading comprehension model.
They have the following problems:

- reading comprehension doesn't perform well at large scale retrieval(uses cross attention, which is slow).
- Systems that first fetch the documents and then search for answers in those docs risk missing good answers from documents that appear to have less relevance to the question.

Because of these problems there is growing interest in training end-to-end retrieval systems where a neural model encodes questions and answers independently as high dimensional vectors, and the relevance of a QA pair is computed by taking their dot product. Retrieval of relevant answers is done using approximate nearest neighbor search to find the answer vectors closest to the question vector which has a complexity of log(N) where N is the number of documents.

A retrieval model should be context aware, since even though the answer to a question is just one sentence the context will help the model find the most relevant sentences, that's why when encoding answers they encourage encoding also the context with it.

Diverse Models were tested in REQA, the best performing one was USE-QA[9]: end-to-end, Questions and answers are encoded independently using a 6 layer transformer encoder and then reduced to a fixed-length vector through average pooling.

A BERT baseline for natural questions

In this paper a new state of the art F1 score is defined on QA trained and evaluated on NQ dataset. When comparing with documentQA[5] baseline the F1 score went from 46.1 to 64.7 and from 35.7 to 52.7 for the long and short answers respectively. Single human baselines for the same tasks are 73.4 and 57.5.

NQ presents a harder challenge than SQUAD 2.0 and CoQA, and the authors think it's the more challenging QA dataset to date because:
Questions in NQ were formulated out of need. In other datasets people got paid for creating questions given a document.
The questions were formulated by people before they had seen the document that would contain the answer
The documents where answers are found are much longer than in other datasets

The model uses BERT to create encoding of documents and answers.
In order to train the model for every document a sliding window is used to break it up in subsets of 128 tokens each. Each training instance will then have 512 tokens with the following format: START_TOKEN + QUESTION_TOKENS + SEPARATOR_TOKEN + DOC_SUBSET_128 + SEPARATOR_TOKEN. Since each question can have long and short answers for every instance the shortest possible answer which is contained in the 128 token subdoc is selected, if no answer available the start and end tokens will point to the START_TOKEN.
Each training set is a tuple of the form: (c,s,e,t), c is a vector of 512 word ids(question, document tokens, markup). s and e are indices pointing to the start and end of the target answer span and t is the answer type(short, long, yes,no, no-answer)

$\text{Loss} = -\log p(s,e,t \mid C) = -\log p_{start}(s|c) - \log p_{end}(e|c) - \log p_{type}(t|c)$
P is obtained as a softmax over scores computed by the BERT model.

At inference time we use our model to obtain a vector representation for the question and for each 128 token span in our corpora, today's most used technique is to use the approximate nearest neighbor algorithm to find the closest answer vector to the vector that represents the question.

Compare and Contrast

The paper "Attention is all you need" introduces the transformer architecture which is leveraged by almost all new models that get SOTA results for information retrieval tasks. This paper used

the transformer to solve the machine translation task. Roberta on the other hand uses that architecture, and a large amount of training data with different training objectives to pre-train a language model which gives SOTA results on several nlp tasks after fine tuning. The paper BERT baseline for Natural questions[4] applies the BERT model to the NQ dataset which is a harder dataset, with real questions and longer documents that may contain their answers. They formulate the loss function and define the training data in a way that allows them to jointly predict the short and long answers using a single BERT model.

The paper on conversational machine comprehension[13] adapts BERT for multi-turn questions in a dialogue system. They propose an ingenious method to model the passage, current question and question history from a dialogue, as input to BERT.

The Squash system for generating question-answer hierarchies[10] is another interesting paper which can potentially be used to generate labelled data for training dialogue or qna systems for information extraction from a document corpus e.g. covid-19 dataset. One of the shortcomings of this system is that the answers cannot span multiple sentences and questions generated cannot span multiple paragraphs.

Future Work

The typical end-to-end conversational or QA systems includes the following critical components,

1. Define the problem or domain in mathematical terms.
2. Transform the objectives of the problem into attainable NLP tasks and then construct training and testing sets accordingly.
3. Choose a model framework to work with and select one of the classic or state-of-the-art model architecture as a basis.
4. Either build a novel model if time and resources are allowed or leveraging the powerful open source pre-train model and then fine-turning on the down-stream tasks.
5. Choose one of the open-source datasets to test your model and evaluate its performance via pre-defined NLP metrics.
6. A few extra steps might be needed for different use cases.

Each step is equally important to make a viable breakthrough on the new system. And after accessing a list of different pre-train models, training datasets, performance metrics, embedding mechanisms, training schemes, etc. The team decided to focus on two areas that would be potentially beneficial to the NLP community as a whole.

The first area is to explore the possibility of a light-weight implementation of the existing model architecture. Most of the existing powerful model stacks numerous layers to deepen the network. This leads to the problem of (i) hard for the industry to fine-tune to model on a small or specific

domain, (ii) hard to re-train the model giving the fact that those models require extensive computing power and data source, (iii) inflexible for the industry to custom the model because of the perplexity of the model structure, and (iv) hard to interpret the result and drill down on the problems.

The second area is to seek for a standard and consistent way to produce a labeled training set from any given document. For example, for a given text corpus, generate viable questions - answers pairs to train your chat system. Or to help build logical hierarchy for any given questions, which will enable people to train a dialog system that can answer a sequence of questions.

The above are just two hunch from our initial research and investigation. The topics might change slightly or deviate according to the follow-up studies on access to the resources and timeline constraints.

References

1. Amin Ahmad, Noah Constant, Yinfei Yang, Daniel Cer. 2019. ReQA: an evaluation for end to end answer retrieval models. arxiv
2. Anna Rogers, Olga Kovaleva, Anna Rumshisky. 2020. A Primer in BERTology: What we know about how BERT works. Arxiv
3. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. 2017. Attention Is All You Need. Arxiv
4. Chris Alberti, Kenton Lee, Michael Collins. 2019. A BERT baseline for natural questions. arxiv
5. Christopher Clark and Matt Gardner. 2017. Simple and effective multi-paragraph reading comprehension. arXiv
6. Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, Bill Dolan. 2007. The third PASCAL recognizing Textual Entailment Challenge. aclweb
7. Geoffrey Hinton, Oriol Vinyals, Jeff Dean. Distilling the Knowledge in a Neural Network. 2015. Arxiv
8. Iulia Turc, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. 2019. Well-Read Students Learn Better: On the Importance of Pre-training Compact Models. arxiv
9. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Arxiv
10. Kalpesh Krishna, Mohit Iyyer. 2019. Generating Question-Answer Hierarchies. Arxiv
11. Thomas Wolf, Victor Sanh, Julien Chaumond, Clement Delangue. 2019. TransferTransfo: A Transfer Learning Approach for Neural Network Based Conversational Agents. Arxiv
12. Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, Qun Liu. 2019. TinyBERT: Distilling BERT for Natural Language Understanding. arxiv
13. Yasuhito Ohsugi, Itsumi Saito, Kyosuke Nishida, Hisako Asano, Junji Tomita. 2019. A Simple but Effective Method to Incorporate Multi-turn Context with BERT for Conversational Machine Comprehension. Arxiv
14. Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, Ray Kurzweil. 2019. Multilingual Universal Sentence Encoder for Semantic Retrieval. Arxiv
15. Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. Arxiv