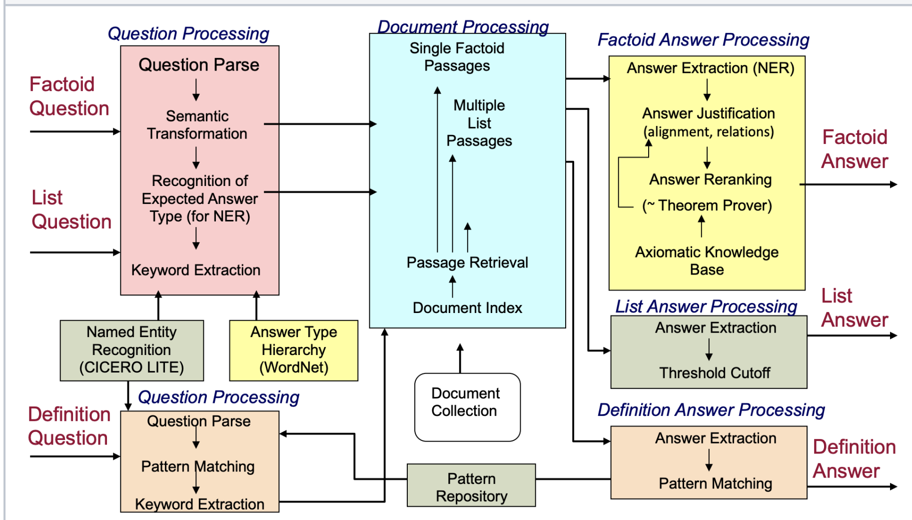
Question-Answering

> ... *reviewing machine learning literature about question answering*

- reference
  - stanford lecture in 2019
  - https://youtu.be/yIdF-17HwSk
  - https://web.stanford.edu/class/cs224n/slides/cs224n-2019-lecture10-QA.pdf
- QA (Question Answering)
  - is a computer science discipline within the fields of information retrieval and natural language processing (NLP)
  - which is concerned with building systems that automatically answer questions posed by humans in a natural language
  - two parts
    - finding document that contains an answer traditional information retrieval or web search
    - finding an answer in a paragraph or a document reading comprehension
- dataset
  - before 2015,
    - MCTest (Machine Comprehension Test) : 2600 questions
    - ProcessBank (describe biological processes) : 500 questions
  - since 2015,
    - CNN/Daily Mail - news stories in CNN and Daily Mail websites as questions
    - SQuAD - Stanford Question Answering Dataset, mostly used
    - LAMBADA - LAnguage Modelling Broadened to Account for Discourse Aspects, narrative passage
    - WDW - Who did What, Toyota Technological Institute at Chicago, a large-scale person centered dataset
    - CBT - Children's Book Test, facebook
    - MS MARCO - MS, Bing questions and human generated answer dataset
    - NewsQA - Maluuba, MS research
    - TriviaQA - trivia enthusiasts and independently gathered evidence documents, 650K question-answer-evidence triples
    - RACE- Carnegie Mellon University, 28,000 passages and 100,000 questions
    - SearchQA- NYU, dataset from IBM's jeopardy archives, consists of more than 140k question-answer pairs
- early work

<table>
<tr><th>early NLP work</th><th>description</th></tr>
<tr><td>


</td><td>
architecture of LCC QA System, 2003

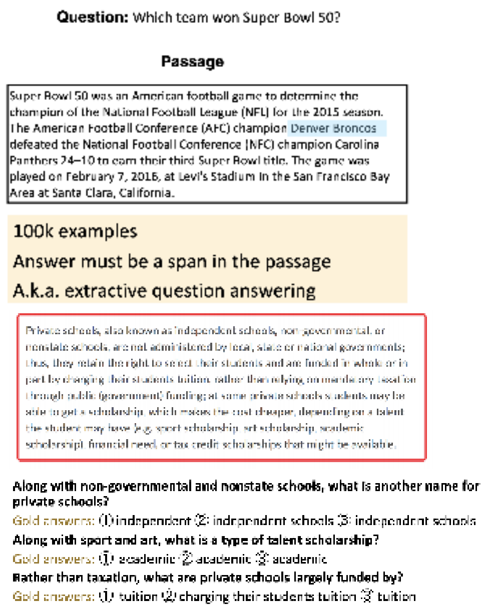
https://trec.nist.gov/pubs/trec11/papers/lcc.moldovan.pdf
complex system with many components
mainly based on NER
work fairly well on factoid questions
factoid questions

questions for which a complete answer can be given in 50 bytes or less,
which is roughly a few words
answer will be semantic simply entity

</td></tr>
</table>

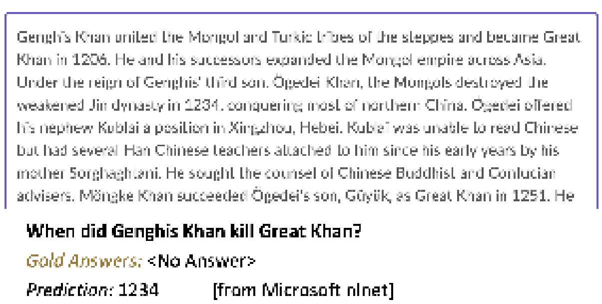
- SQuAD (Stanford Question Answering Dataset)

SQuAD	description



v1.1

- authors collected 3 gold answers
  - robust to variation in human's answers
- systems are scored on two metrics
  - exact match and F1 score
- ignore punctuation and articles
- https://rajpurkar.github.io/SQuAD-explorer/



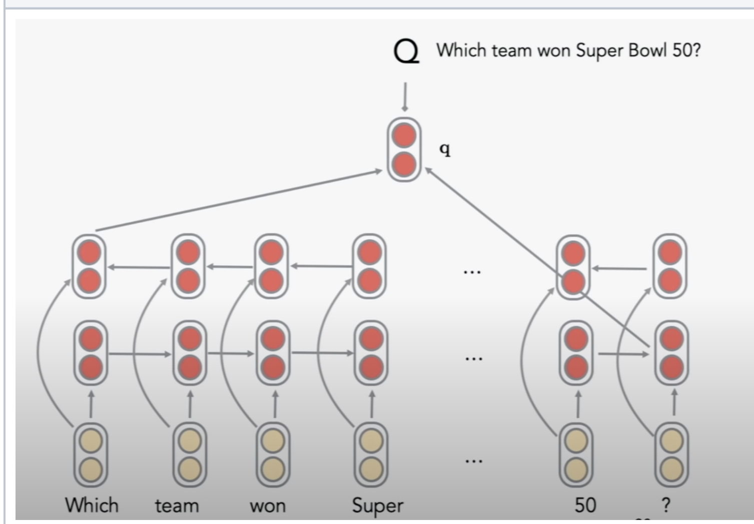
v2.0

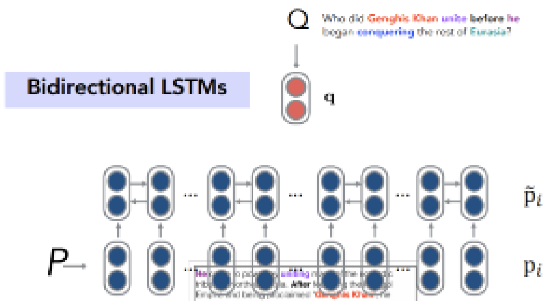
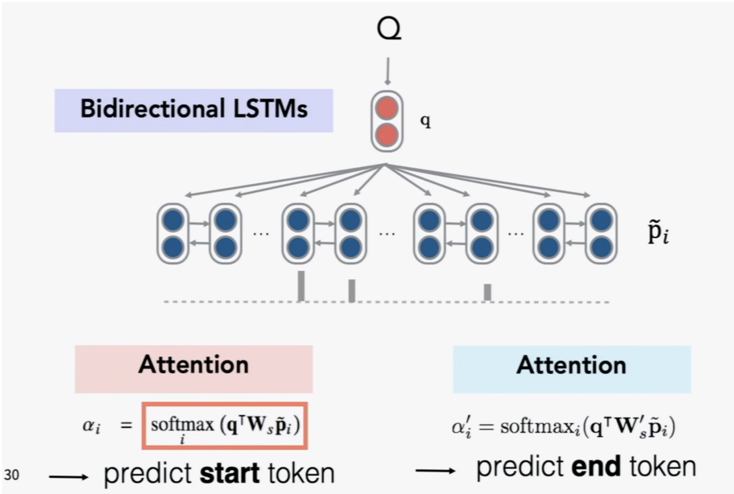
- a defect of v1.1 is that all questions have an answer in the paragraph
  - system rank candidates and choose the best one
  - without understanding context
- have no answer question in 1/3 of the training and 1/2 of the dev/test questions
  - have a threshold score for whether a span answers a question

limitation of SQuAD

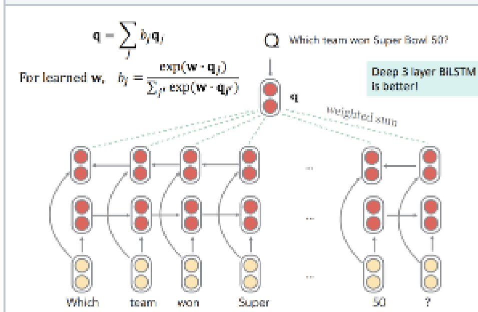
- well-targeted, well-structured and clean dataset but,
- only span-based answers (no yes/no, counting, implicit why)
- questions were constructed looking at the passages
  - not genuine information needs
- barely any multi-fact/sentence inference beyond coreference
- https://arxiv.org/pdf/1806.03822.pdf

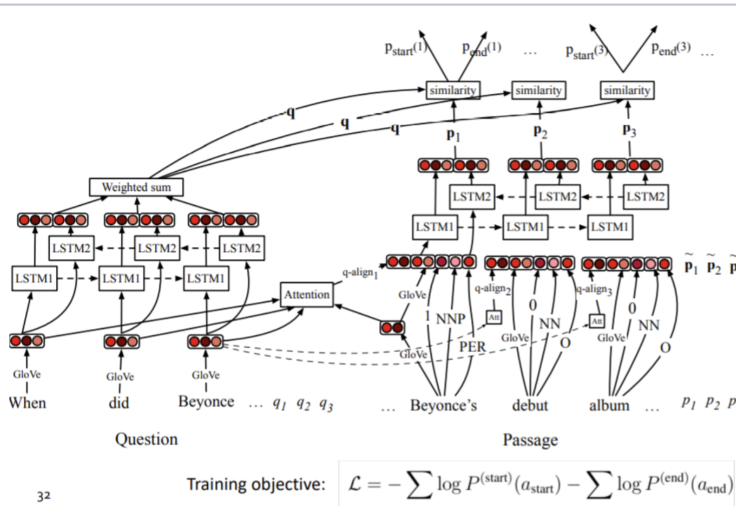
- a family of LSTM-based models with attention (2016-2018)
  - Stanford Attentive Reader

Stanford Attentive Reader	description
	• simplest neural question answering system • Bi-LSTM • DrQA - 2016, uses IR(information retrieval) followed by neural reading comprehension to bring deep learning to open-domain QA 1. embedding of question • word embedding for each word (GloVe 300d) • run LSTM forward, and Bi-LSTM backward of the question and get two end hidden state • concatenate two end state to one 2-dim vector • that is a representation of a question

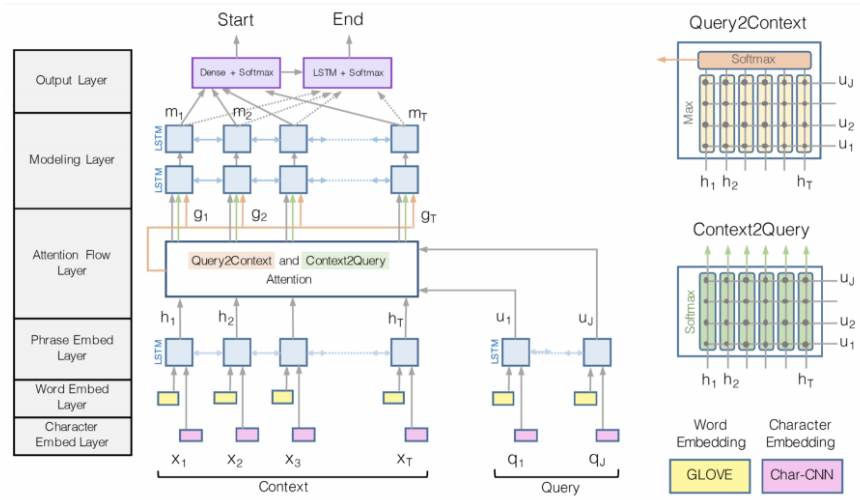
	2. embedding of passage ▪ word vector for all words of a passage (Pi) ▪ Bi-LSTM for each word embedding which imply passage, and get the LSTM representation (~Pi)
	3. find the answer using Bi-linear attention ▪ get the attention score using bi-linear attention ▪ predict start token using attention score ▪ predict end token using attention score

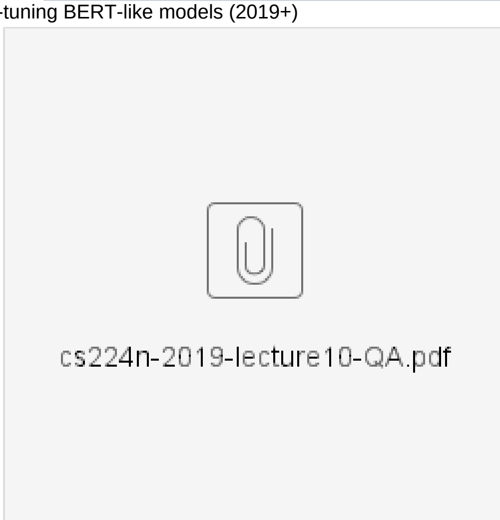
- Stanford Attentive Reader ++

Stanford Attentive Reader++	difference from previous version of model
	1. end states all states 2. Deep 3 layer BiLSTM which is better than single layer

	3. passage representation
	▪ aligned question embedding ('car' and 'vehicle')
	  ▪ similarity score
	  ▪ to better understanding between question and passage

- BiDAF (Bi-Directional Attention Flow for Machine Comprehension)

BiDAF	description
	▪ https://arxiv.org/pdf/1611.01603.pdf ▪ character level embedding ▪ attention flow layer works bi-directional between query to context ▪ attention should flow both ways from the context to the question and from the question to the context ▪ based on similarity score for each passage word and question word ▪ Context2Query : which word in query is the most relevant to context word? ▪ Query2Context : which word in context is the most relevant to query word?

- fine-tuning BERT-like models (2019+)
- ref: 

DISFL-QA: A Benchmark Dataset for Understanding Disfluencies in Question Answering

- Goal : presents a new challenge question answering dataset, Disfl-QA, a derivative of SQuAD, where humans introduce contextual disfluencies in previously fluent questions.
  - Link : https://research.google/pubs/pub50373/ (06/2021)
  - Dataset : https://github.com/google-research-datasets/disfl-qa
  - Credible source: Google Research, Association for Computational Linguistics

Background knowledge

- disfluencies
  - a natural conversation often includes interruptions like repetitions, restarts, or corrections
    - these phenomena are referred to as disfluencies
    - an NLU(Natural Language Understanding) system, trained on fluent data, can easily get misled due to their presence
  - SQuAD(Stanford Question Answering Dataset)
    - a dataset for reading comprehension
    - consists of a list of questions by crowdworkers on a set of Wikipedia articles
    - the answers to each of the questions is a segment of text, or span, from the corresponding Wikipedia reading passage
    - SQuAD 1.1 consists of 100,000+ question and answer pairs on 500+ articles.
    - SQuAD2.0 combines the 100k questions from its predecessor, SQuAD1.1, with 50k+ additional unanswerable questions from crowdworkers.
    - https://huggingface.co/datasets/squad_v2

- example

```
{
    "answers": {
        "answer_start": [94, 87, 94, 94],
        "text": ["10th and 11th centuries", "in the 10th and 11th centuries", "10th and 11th
centuries", "10th and 11th centuries"]
    },
    "context": "\"The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the
people who in the 10th and 11th centuries gave thei...",
    "id": "56ddde6b9a695914005b9629",
    "question": "When were the Normans in Normandy?",
    "title": "Normans"
}
```


Build Disfl-QA dataset

1. annotation task
    a. first round of annotation - the expert rater provide a disfluent version of the question which should be
        - semantically equivalent to the original question
            - natural, i.e., a human can utter them in a dialogue setting
            - not include partial words of filled pauses
        - Human Evaluation + Re-annotation - to ensure the quality of the dataset, let another set of human raters assess
            - if the disfluent question consistent with respect to the fluent question
            - if the disfluent question natural
            - for the cases identified as either inconsistent or unnatural, conducts a second round of re-annotation with updated guidelines to make required corrections
    b. assess the distribution of disfluencies
        i. sampled 500 questions and manually annotated the nature of disfluency
        ii. compare with SWITCHBOARD data set(https://ieeexplore.ieee.org/document/225858)
            1. SWITCHBOARD dataset is a large multi speaker corpus of conversational speech
            2. DISFL-QA is more diverse, contains a larger fraction of corrections and restarts
            3. DISFL-QA also contains more coreference

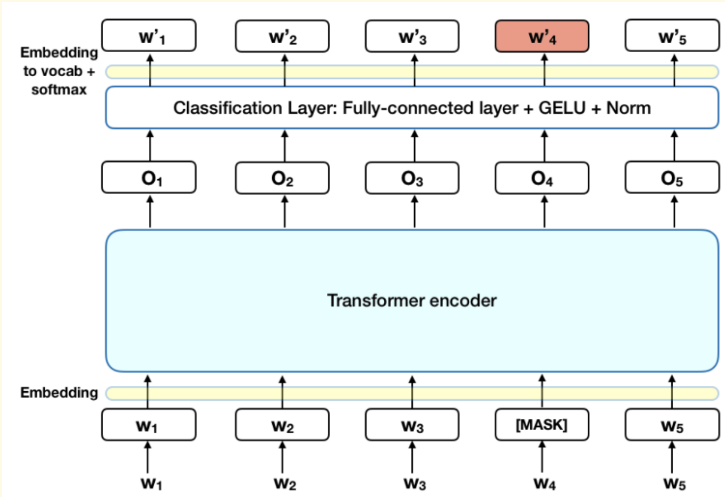
Model

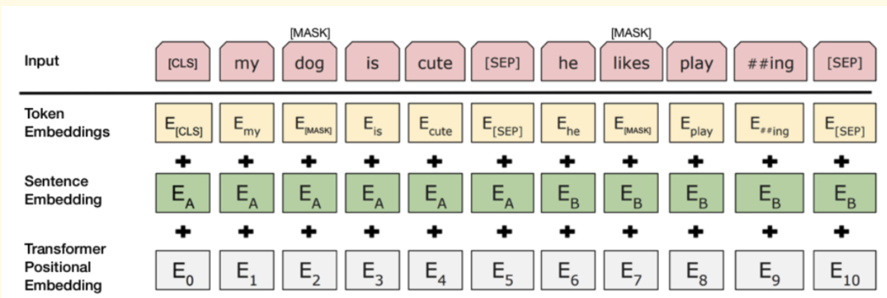
Transformer

- as opposed to directional models, which read the text input sequentially, the Transformer encoder reads the entire sequence of words at once
    - it is considered bidirectional
    - allows the model to learn the context of a word based on all of its surroundings

BERT (Bidirectional Encoder Representations from Transformers)

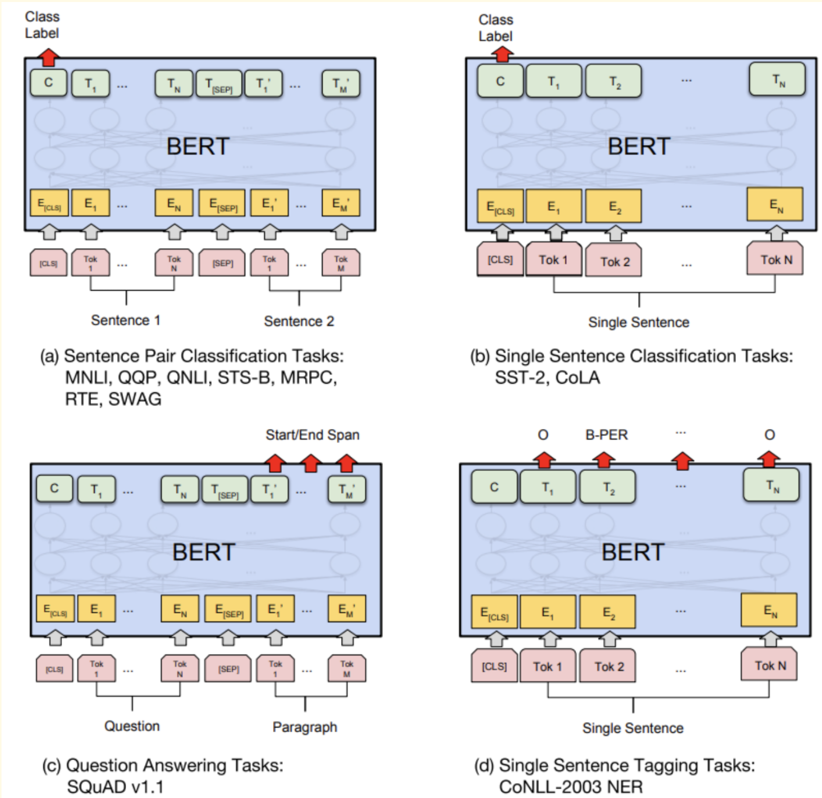
- use of transformer, an attention mechanism that learns contextual relations between words in a text
    - BERT uses two training strategies

<table>
<tr><th>training strategies in BERT</th><th>description</th></tr>
<tr><td>
</td><td>Masked Language Model (MLM)

• before feeding word sequences into BERT, 15% of the words in each sequence are replaced with a MASK token
    • then attempts to predict the value of the masked words based on the context provided by non masked words</td></tr>
<tr><td>
</td><td>Next Sentence Prediction (NSP)

• model receives pairs of sentences as input
    • and learns to predict if the second sentence in the pair is the subsequent sentence in the original document
    • during the training 50% of the inputs are a pair in which the second sentence is the subsequent sentence, while other 50% a random sentence from the corpus
    • To do that, the input is processed in 3 embeddings
        • Token Embeddings
            • [CLS] token is inserted at the beginning of the first sentence
            • [SEP] token is inserted at the end of each sentence
        • Sentence Embeddings
            • indicates sentence,
            • added to each token
        • Positional Embeddings
            • added to each token to indicate its position in the sequence
    • then the output of the [CLS] token is transformed into a 2*1 vector,
    • and finally calculate the probability of IsNextSequence with softmax</td></tr>
</table>

<table>
<tr><th>transfer learning in BERT</th><th>description</th></tr>
<tr><td>



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

(c) Question Answering Tasks: SQuAD v1.1

(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

</td><td>

Transfer learning

- model is first pre-trained on a data-rich task
  - then fine tuned on a downstream tasks
  - https://ai.googleblog.com/2020/02/exploring-transfer-learning-with-t5.html

How to use BERT (Fine-tuning)

- BERT can be used for a wide variety of language tasks, while only adding a small layer to the core model:

1. Classification tasks such as sentiment analysis are done similarly to Next Sentence classification, by adding a classification layer on top of the Transformer output for the [CLS] token.
   1. In Question Answering tasks (e.g. SQuAD v1.1), the software receives a question regarding a text sequence and is required to mark the answer in the sequence. Using BERT, a Q&A model can be trained by learning two extra vectors that mark the beginning and the end of the answer.
   2. In Named Entity Recognition (NER), the software receives a text sequence and is required to mark the various types of entities (Person, Organisation, Date, etc) that appear in the text. Using BERT, a NER model can be trained by feeding the output vector of each token into a classification layer that predicts the NER label.

</td></tr>
</table>

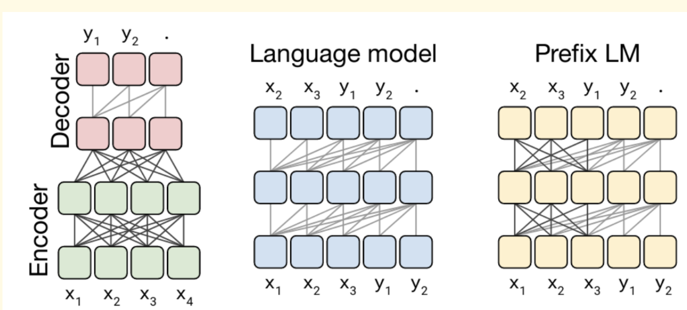
- https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

T5 (Text-To-Text Transfer Transformer)

- developed by Google
  - presented in https://arxiv.org/pdf/1910.10683.pdf
  - code is available in https://github.com/google-research/text-to-text-transfer-transformer
  - T5 model, pre-trained on C4
    - C4 : open-source pre-training dataset, called the Colossal Clean Crawled Corpus(C4) which is web crawl corpus (https://commoncrawl.org)
  - achieves state-of-the-art results on many NLP benchmarks
  - With T5, we propose reframing all NLP tasks into a unified text-to-text-format where the input and output are always text strings
  - in contrast to BERT-style models that can only output either a class label or a span of the input

<table>
<tr><th>training approach in T5</th><th>description</th></tr>
<tr><td>

model structures



</td><td>

1. encoder-decoder : standard encoder-decoder
   1. language model : the causal attention mechanism
   2. prefix LM : combination of the BERT-style and language model approaches

</td></tr>
</table>

  - https://towardsdatascience.com/t5-text-to-text-transfer-transformer-643f89e8905e

Experiment

training data

Human Annotated Datasets

- SQuAD
  - DISFL-QA
    - split the 11,825 annotated questions in DISFL-QA into train/dev/test set containing 7,182/1,000/3, 643 questions

Heuristically Generated Data

- generate disfluencies heuristically to validate the importance of human annotated disfluencies
  - SWITCH-Q : insert prefix of another question as a prefix to the original question
  - SWITCH-X : X could be verb, adjective, adverb, or entity, and insert as a reparandum in the question

experiment

Modelling approaches

1. LMs for QA
   - BERT
     - fine tuned BERT for a span selection task which is
       - predicting start and end probabilities for all the tokens in the context

evaluation and output

- evaluate QA performance using the standard SQuAD-v2 evaluation script which report EM and F1 scores over the HasAns(answerable) and NoAns(non-answerable)
  - BERT-QA and T5-QA models
    - trained only the SQuAD dataset and
    - evaluated on SQuAD, Heuristics, and DISFL-QA test sets
  - Disfluency Correction + T5-QA
    - BERT based disfluency correction as a preprocessing step
    - feed the input to T5-QA model
    - trained on SWITCHBOARD dataset

Model	Train	Eval	HasAns-F1	NoAns-F1	Overall-F1
BERT-QA	ALL	SQuAD	83.87	70.55	77.46
		Heuristics	51.45 ↓ 32.42	74.49 ↑ 3.94	62.53 ↓ 14.93
		DISFL-QA	40.97 ↓ 42.90	75.97 ↑ 5.42	57.81 ↓ 19.65
	ANS	SQuAD	89.63	-	89.63
		Heuristics	80.52 ↓ 9.11	-	80.52 ↓ 9.11
		DISFL-QA	78.88 ↓ 10.75	-	78.88 ↓ 10.75
T5-QA	ALL	SQuAD	91.38	87.67	89.59
		Heuristics	39.98 ↓ 51.40	92.57 ↑ 4.90	65.27 ↓ 24.32
		DISFL-QA	35.31 ↓ 56.07	90.06 ↑ 2.39	61.64 ↓ 27.95
	ANS	SQuAD	93.71	-	93.71
		Heuristics	81.73 ↓ 12.01	-	81.73 ↓ 12.01
		DISFL-QA	80.39 ↓ 13.32	-	80.39 ↓ 13.32
Disfluency Correction + T5-QA	ALL	SQuAD	91.38	87.67	89.59
		Heuristics	42.83 ↓ 48.55	92.18 ↑ 4.51	66.56 ↓ 23.03
		DISFL-QA	43.61 ↓ 47.77	89.55 ↑ 1.88	65.71 ↓ 23.88
	ANS	SQuAD	93.71	-	93.71
		Heuristics	82.27 ↓ 10.44	-	82.27 ↓ 10.44
		DISFL-QA	82.64 ↓ 11.07	-	82.64 ↓ 11.07

Result

- on heuristics and DISFL-QA test sets, both the BERT-QA and T5-QA models exhibit significant performance drop
  - BERT and T5 are not robust when questions contain disfluencies
  - in general, DISFL-QA exhibit larger performance drop compared to heuristics across different models
  - T5-ALL shows that DISF-QA shows a bigger drop in HasAns, smaller increase in NoAns compare Heuristics test set
  - T5-ANS shows that DISF-QA shows a larger drop in performance which is attributed to the model picking wrong answer span
  - hypothesis : heuristics are to confuse the models in over-predicting <no answer>, but DISFL-QA is superior when it comes to confuse the models to picking a different answer span altogether

collecting a dataset like DISFL-QA via human annotation holds value for contextual disfluencies

- performance drop is largely due to the drop in F1 on HasAns questions than NoAns quations
  - major fraction of prediction errors for HasAns is attributed to HasAns NoAns error, instead of HasAns WrongAns

the disfluencies are causing the answerable questions to resemble the non-answerable ones

	- T5
	  - fine tuned under the standard text2text formulation
	  - when given (question, passage) as input the model generates the answer as the output
	  - for prediction (no answer), the model was trained to generate 'unknown'	

- LMs for disfluency correction
  - given the disfluent question as input
  - a correction model predicts the fluent question
  - then fed into a QA model
  - BERT
    - BERT-based disfluency correction model trained on SWITCHBOARD
  - T5
    - T5 model trained on DISFL-QA
    - to prevent the distribution skew between SWITCHBOARD and DISFL-QA

Training setting

	• ALL - model is trained on all of SQuAD-v2 including the non answerable questions ◦ ANS - model is trained only on answerable questions from SQuAD-v1	

Data

"5a5918ff3e1742001a15cf7e": {"original": "What do unstable isotope studies indicate?", "disfluent": "What do petrologists no what do unstable isotope studies indicate?"},
"5ad4f40c5b96ef001a10a774": {"original": "What is the basic unit of territorial division in Warsaw?", "disfluent": "What is the second level of territorial division in Poland no make that the basic unit of territorial division in Warsaw?"},
"572684365951b619008f7543": {"original": "Which genus lack tentacles and sheaths?", "disfluent": "Juvenile platyctenids no wow Which genus lack tentacles and sheaths?"},
"5729f799af94a219006aa70a": {"original": "Long-lived memory cells can remember previous encounters with what?", "disfluent": "When a pathogen is met again scratch that I mean long-lived memory cells are capable of remembering previous encounters with what?"},
"5ad3b9cd604f3c001a3fee87": {"original": "What led to Newcastle's rise to power as military advisor?", "disfluent": "What led to the Duke of Cumberland's rise to power as military advisor sorry no Newcastle's?"},
"5a665b56846392001a1e1b1d": {"original": "How long did Julia Butterfly Hill live near a nuclear-missile installation?", "disfluent": "Did Julia Butterfly wait How long did Julia Butterfly Hill live near a nuclear-missile installation?"},

BERT base

question and answering task #1	**question and answering task #2**

Hugging Face Transformer

- open-source provider of natural language processing (NLP) technologies
- transformer is one of NLP library
- transformers library has a lot of different BERT models
- model name : bert-large-uncased-whole-word-masking-finetuned-squad
- https://huggingface.co/

BERT

- Bidirectional Encoder Representations from Transformers (BERT) is one of the most popular and widely used NLP models
- useful for understanding the intent behind the query asked
- tokeniser
    - [CLS] token stands for classification and is there to represent sentence-level classification.
    - [SEP] is used to separate the two pieces of text.
- wordpiece tokenisation
    - subword tokenisation algorithms used for BERT
    - wordpiece tokenisation uses ## to delimit tokens that have been split
    - the idea behind this is to reduce the size of the vocabulary which improves training performance
    - i.e. Consider the words, run, running, runner. each of the three words would be split into 'run' and the related '##SUFFIX' (if any suffix at all — for example, "run", "##ning", "##ner"). Now, the model will learn the context of the word "run" and the rest of the meaning would be encoded in the suffix, which would be learned from other words with similar suffixes

CoQA Dataset for test

- Conversational Question Answering (CoQA) dataset is released by Stanford NLP in 2019
- large-scale dataset for building Conversational Question Answering Systems
- aims to measure the ability of machines to understand a text passage and answer a series of interconnected questions that appear in a conversation
- `http://downloads.cs.stanford.edu/nlp/data/coqa/coqa-train-v1.0.json`

Hugging Face Transformer

- model name : deepset/bert-base-cased-squad2
- pipeline()
    - offering a simple API dedicated to several tasks, including Named Entity Recognition, Masked Language Modelling, Sentiment Analysis, Feature Extraction and Question Answering
    - https://huggingface.co/transformers/main_classes/pipelines.html

BERT

- tokens

Token	Meaning	Token ID
[PAD]	Padding token, allows us to maintain same-length sequences (512 tokens for Bert) even when different sized sentences are fed in	0
[UNK]	Used when a word is unknown to Bert	100
[CLS]	Appears at the start of every sequence	101
[SEP]	Indicates a seperator - used to indicate point between context-question and appears at end of sequences	102
[MASK]	Used when masking tokens, for example in training with masked language modelling (MLM)	103

ref. https://towardsdatascience.com/question-answering-with-a-fine-tuned-bert-bc4dafd45626

ref. https://towardsdatascience.com/question-and-answering-with-bert-6ef89a78dac

BERT fine-tuned

text classification task	sentiment analysis task

BERT fine-tuning

- BERT is a big neural network architecture, with a huge number of parameters, that can range from 100 million to over 300 million.
- Training a BERT model from scratch on a small dataset would result in overfitting.
- It is better to use a pre-trained BERT model that was trained on a huge dataset, as a starting point.
- We can then further train the model on our relatively smaller dataset and this process is known as model fine-tuning.

Different fine-tuning techniques

- **Train the entire architecture** – We can further train the entire pre-trained model on our dataset and feed the output to a softmax layer. In this case, the error is back-propagated through the entire architecture and the pre-trained weights of the model are updated based on the new dataset.
- **Train some layers while freezing others** – Another way to use a pre-trained model is to train it partially. What we can do is keep the weights of initial layers of the model frozen while we retrain only the higher layers. We can try and test as to how many layers to be frozen and how many to be trained.
- **Freeze the entire architecture** – We can even freeze all the layers of the model and attach a few neural network layers of our own and train this new model. Note that the weights of only the attached layers will be updated during model training.

Dataset

- https://raw.githubusercontent.com/prateekjoshi565/Fine-Tuning-BERT/master/spamdata_v2.csv
- The dataset consists of two columns – “label” and “text”. The column “text” contains the message body and the “label” is a binary variable where 1 means spam and 0 means the message is not a spam.

Hugging Face Transformer

- model name : bert-base-uncased (110 million parameters)
- padding
  - the text in the dataset are of varying length
  - use padding to make all the messages have the same length
  - if we set the maximum length of text as padding length, it will make the training slower

Dataset

- https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews
- IMDB dataset having 50K movie reviews for natural language processing or Text analytics.
- The dataset consists of two columns - "review" and "sentiment". The column "review" contains the text and the "review" is a binary variable either "positive" or "negative".

Hugging Face Transformer

- model name : bert-base-multilingual-cased

CPU vs GPU vs TPU

- CPU handles all the logics, calculations, and input/output of the computer, it is a general-purpose processor. In comparison, GPU is an additional processor to enhance the graphical interface and run high-end tasks. TPUs are powerful custom-built processors to run the project made on a specific framework, i.e. TensorFlow.
  - **CPU:** Central Processing Unit. Manage all the functions of a computer.
  - **GPU:** Graphical Processing Unit. Enhance the graphical performance of the computer.
  - **TPU:** Tensor Processing Unit. Custom build ASIC(Application Specific Integrated Circuit) to accelerate TensorFlow projects.

ref. https://www.analyticsvidhya.com/blog/2020/07/transfer-learning-for-nlp-fine-tuning-bert-for-text-classification/

https://github.com/prateekjoshi565/Fine-Tuning-BERT/blob/master/Fine_Tuning_BERT_for_Spam_Classification.ipynb

https://pypi.org/project/keras-bert/, https://github.com/CyberZHG/keras-bert/tree/master/keras_bert