Post-BERT Era

Tasks as text-to-text problem

Learning goals

- reformulating classification tasks
- multi-task learning
- fine-tuning on task-prefixes

RECAP: BERT, ROBERTA, ETC.

- Transformer encoder
- Training: Masked language modeling (or similar)
- BERT* learns an enormous amount of knowledge about language and the world through MLM training on large corpora
- Application: fine-tune on a particular task
- Great performance!
- **Question:** What's not to like?

*In what follows we will use BERT as a representative for this class of language models and only talk about BERT – but the discussion includes RoBERTa, Albert, XLNet etc.

PROBLEMS WITH BERT (1)

- You need a different model for each task.

  (Because BERT is differently fine-tuned for each task.)

  $\rightarrow$ Not realistic in many real deployment scenarios, e.g., on mobile devices.
- *Human learning:* We arguably have a single model that solves all tasks!
- **Question:** Is there a framework that allows us to create a single model that solves all tasks?

PROBLEMS WITH BERT (2)

- Two training modes: first (MLM) pretraining, then fine-tuning.
- Fine-tuning is supervised learning, i.e., learning from labeled examples.
- Arguably, learning from labeled examples is untypical for human learning.
- You never learn a task solely by being presented a bunch of examples, without explanation.
- Instead, in human learning, there is almost always a task description.

PROBLEMS WITH BERT (2)

- Example: *How to boil an egg.*
  - “Place eggs in the bottom of a saucepan.“
  - “Fill the pan with cold water.“
  - “Etc.“
- Notice that this is not an example but a *description* of the task
- **Question:** Is there a framework that allows us to leverage task descriptions?

PROBLEMS WITH BERT (3)

- BERT has great performance, but . . .
- . . . only if the training set is large, generally 1000s of examples
- This is completely different from human learning!
- We do use examples in learning, but in most cases, only a few

PROBLEMS WITH BERT (3)

- Example: Maybe the person teaching you how to boil an egg will show you how to do it *one or two times*
- But probably not 10 times
- Definitely not a 1000 times
- More practical concern: it's very expensive to label 1000s of examples for each task (there are many tasks).
- **Question:** Is there a framework that allows us to learn from just a small number of examples?
- This is called few-shot learning.

PROBLEMS WITH BERT (4)

- More subtle aspect of the same problem (i.e., large training sets):

  $\rightarrow$ **Overfitting**
- Even though performance looks good on standard train/dev/test splits, the deviation between the training set and the data actually encountered in real application can be large
- So our benchmarks often overestimate what performance would be in reality

REVISITING TEXT-TO-TEXT TASKS

Example: Machine Translation

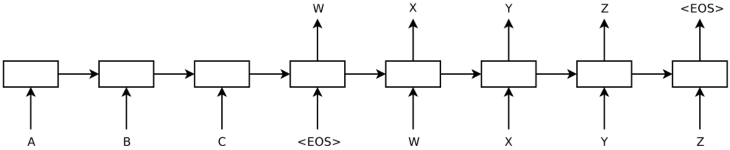
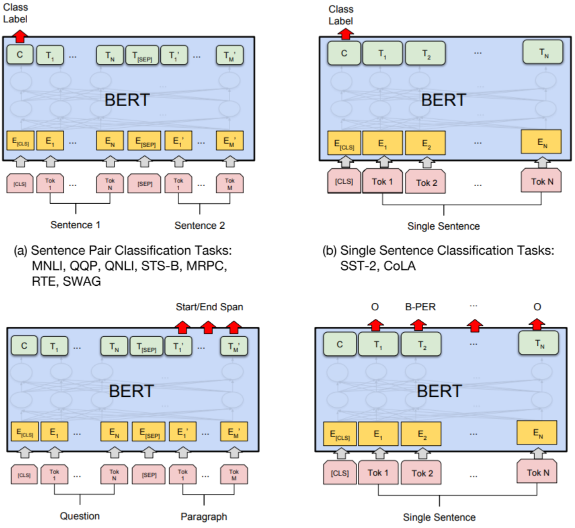


Figure 1: Our model reads an input sentence "ABC" and produces "WXYZ" as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the input sentence in reverse, because doing so introduces many short term dependencies in the data that make the optimization problem much easier.

Source: Sutskever et al., 2014

- *Input:* Text in source language
- *Output:* Text in target language

REVISITING FINE-TUNING OF BERT



(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

Source: Devlin et al., 2019

FINE-TUNING + TEXT-TO-TEXT?

- **Question:** How does BERT know which task it is performing?
  - It doesn't!
  - BERT just learns associations between text features and class distributions
  - No meaningful internal representation of what “class 3“ actually *means*
  - *Input*: text; *Output*: `[0 0 1 0 0 0]`
- The fact that BERT does not “understand“ what task it is actually performing is the reason why one copy is needed per task

FINE-TUNING + TEXT-TO-TEXT?

- **Question:** How could we “inform“ our model about the task?
  - Model takes text as an input and can process.
  - Why not describe the task in natural language?
  - *Assumption:*
    Given the model sees 1000s of (inputs; ouput) pairs like ..

(“`<task description>: <input sample>`“ ; “`<class name>`“)

  - .. it will learn to associate ..
    $\rightarrow$ .. task descriptions to a set of class names
    $\rightarrow$ .. text features to class name distributions

FINE-TUNING + TEXT-TO-TEXT?

- **Question:** How does a “good“ task description look like?
  - Remember: The idea is to have a model that learns to associate the task descriptions with a set of class names
  - *Class names:*
    $\rightarrow$ Could be pretty new/exotic terms the model is not familiar with after pre-training
    $\rightarrow$ It will learn to output them during fine-tuning when “triggered“ accordingly (i.e. with the right task description)
  - *Task descriptions:*
    $\rightarrow$ **Question:** Same intuition here?!

FINE-TUNING + TEXT-TO-TEXT?

- **Question:** If I want the model to do sentiment classification, can I just provide it with 1000s of training samples like

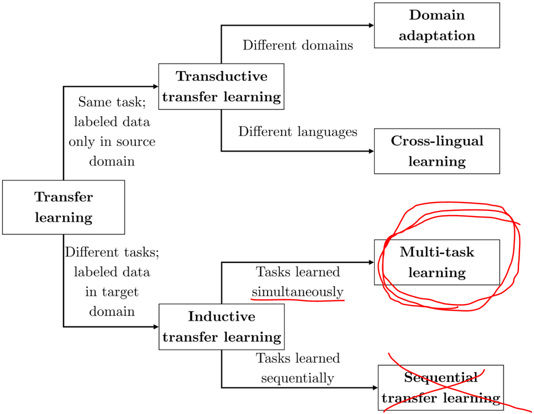
  (“`<khaleesi>: <input sample>`“ ; “`<positive/negative>`“)

  ?

  (*khaleesi** just used as some random fantasy word here)
- *Intuition:* Yes! It basically does not matter, **as long as the model is fine-tuned on this data**
- Fine-tuning important for the model to pick up on this “signal“ and output according text (corresponding the appropriate classes)

*Khaleesi is a Dothraki title referring to the wife of the khal (cf. Game of Thrones).

MULTI-TASK LEARNING?



Taxonomy of transfer learning (Source: Ruder, 2019)

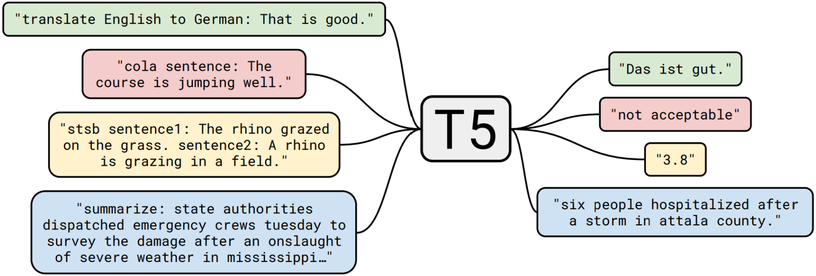
MULTI-TASK LEARNING?

- **Question:** If the model can learn to link task descriptions to label sets, can it learn to perform multiple tasks at once?
- Yes, through fine tuning each task description will be linked to its label set:
  - `<task description a>` $\rightarrow$ `{positive, negative}`
  - `<task description b>` $\rightarrow$ `{world, sports, business, sci/tech}` ▸ Example from AG News
  - `<task description c>` $\rightarrow$ `{spam, no spam}`
  - `<task description d>` $\rightarrow$ `{entailment, contradiction, neutral}`

MULTI-TASK LEARNING?

- **Question:** Can we have one single model for both different classification tasks **and** generation tasks?
  - Yes, we just have to design suitable task descriptions
  - Examples for *generative* descriptions ▸ Raffel et al., 2020
    - `translate English to German: <input sample>`
    - `summarize: <input sample>`
    - `answer question: <input sample>`
  - Examples for *classification* descriptions ▸ Raffel et al., 2020
    - `binary classification: <input sample>`
    - `predict sentiment: <input sample>`

TEXT-TO-TEXT TASKS



Source: Raffel et al., 2020

- **Important note:** What we talked about as `<task description>` until now is commonly referred to as `<task prefix>`