BERT

●●●

An Introduction to BERT and Word Embeddings in general

Anna Wiedenroth

Overview

- What are word embeddings
- ELMo: Introduction of Context
- Transformers: Attention is all you need
- BERT
  - Use cases
  - BERT based Architectures
  - How to get started with BERT
- References: Papers and Blog Articles

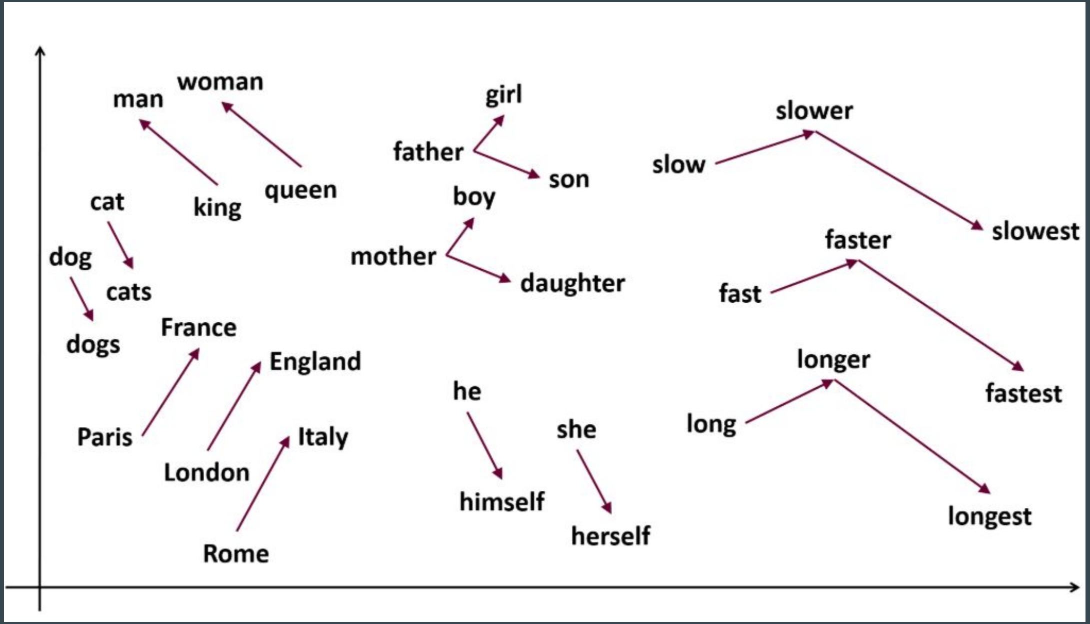
Word Embeddings

Word Embeddings - Intuition

- language needs to be represented in numbers for computer
- first idea one-hot encoding
- ideally these numbers should not be random but reflect the semantics and syntactics, ie. words which are related or used for similar function should be closer together as others
- furthermore we want to do computations on this numbers
- $\rightarrow$ vector space which is not too high dimensional

Word Embeddings - Intuition

“You shall know a word by the company it keeps.” (John Rupert Firth, 1957)



Word Embeddings - Implementation

- simple neural network, only one hidden layer
- train on a “dummy task” i.e. given a word, predict its context
- input are the words as one-hot encodings
- hidden layer works as a lookup table and contains the embeddings of shape (*vocabsize* x *embeddingdim*)
- output layer is only relevant for training
- vocabulary gets projected into space of *embeddingdim* dimensions
- computations possible like “king-man+woman = queen“
- after training on large balanced corpus, embeddings can be reused
- several architectures available, like Word2Vec, GloVe and Fasttext

Word Embeddings - Implementation

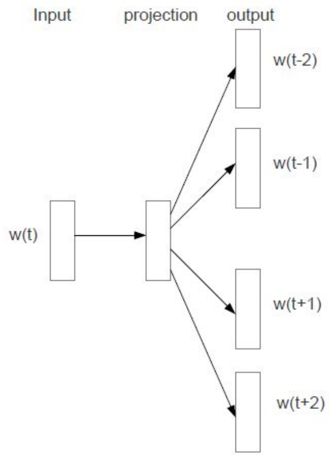


Figure 1: The Skip-gram model architecture. The training objective is to learn word vector representations that are good at predicting the nearby words.

Distributed Representations of Words and Phrases and their Compositionality
by Tomas Mikolov et al. 2013

Word Embeddings - Limitations

- one embedding per word, cannot account for polysemy
- larger structures in language like syntax or long term dependencies cannot be covered
- not directly usable for concrete tasks like e.g. sentiment analysis

⇒ word embeddings are used as input to more complex neural networks

ELMo

ELMo - Intuition

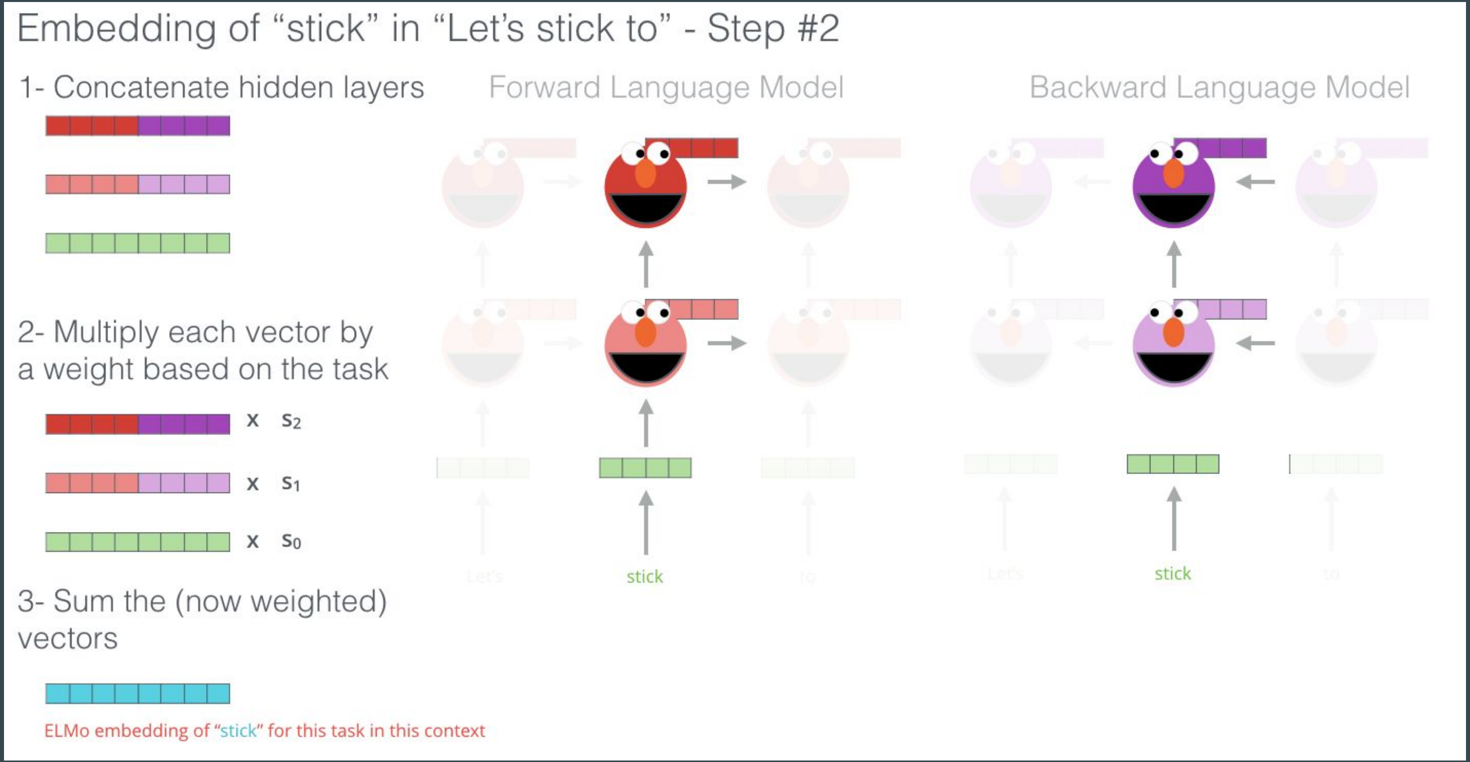
Idea: take not single words as input but full sentences, store context in LSTM

Hey ELMo, what's the embedding of the word "stick"?
There are multiple possible embeddings! Use it in a sentence.
Oh, okay. Here: "Let's stick to improvisation in this skit"
Oh in that case, the embedding is: -0.02, -0.16, 0.12, -0.1 ....etc

ELMo - Implementation

- ELMo stands for “Embeddings from Language Models”
- language models (LM) compute the probability of a sequence of words
  - a forward LM computes probability of token given its predecessors
  - a backward LM predicts token given future context
- LMs are implemented as a certain type of Recurrent Neural Networks, so called LSTMs (Long Short Term Memory cells)
- ELMo concatenates hidden layers of forward and backward LM
- Similar to Word2Vec embeddings, ELMo embeddings are integrated in more complex task specific language model

ELMo - Implementation



Embedding of "stick" in "Let's stick to" - Step #2
1- Concatenate hidden layers
Forward Language Model
Backward Language Model
2- Multiply each vector by a weight based on the task
x s2
x s1
x s0
3- Sum the (now weighted) vectors
ELMo embedding of "stick" for this task in this context
Let's
stick
to
Let's
stick
to

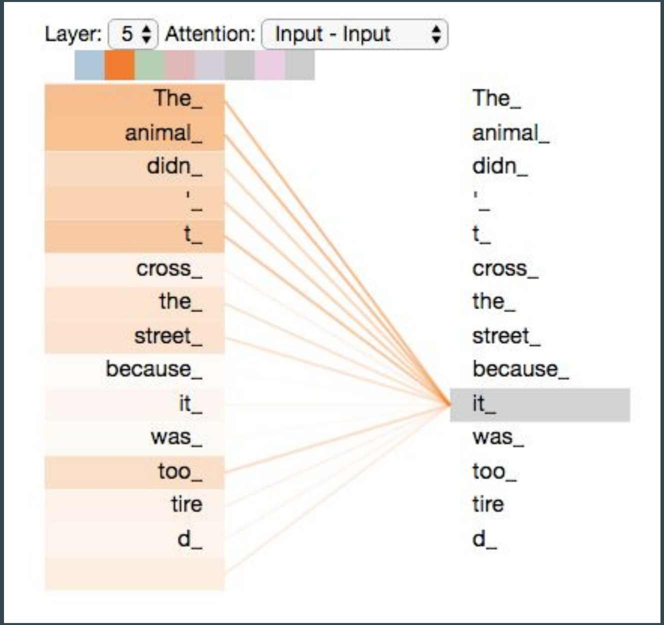
ELMo - Limitations

- LSTMs have problems with long term dependencies - problem if sentences are long
- RNN structure does not allow for parallelization → very high computational cost
- only partial transfer learning: main part of the task specific model still has to be trained from scratch

Transformers

Transformers - Intuition

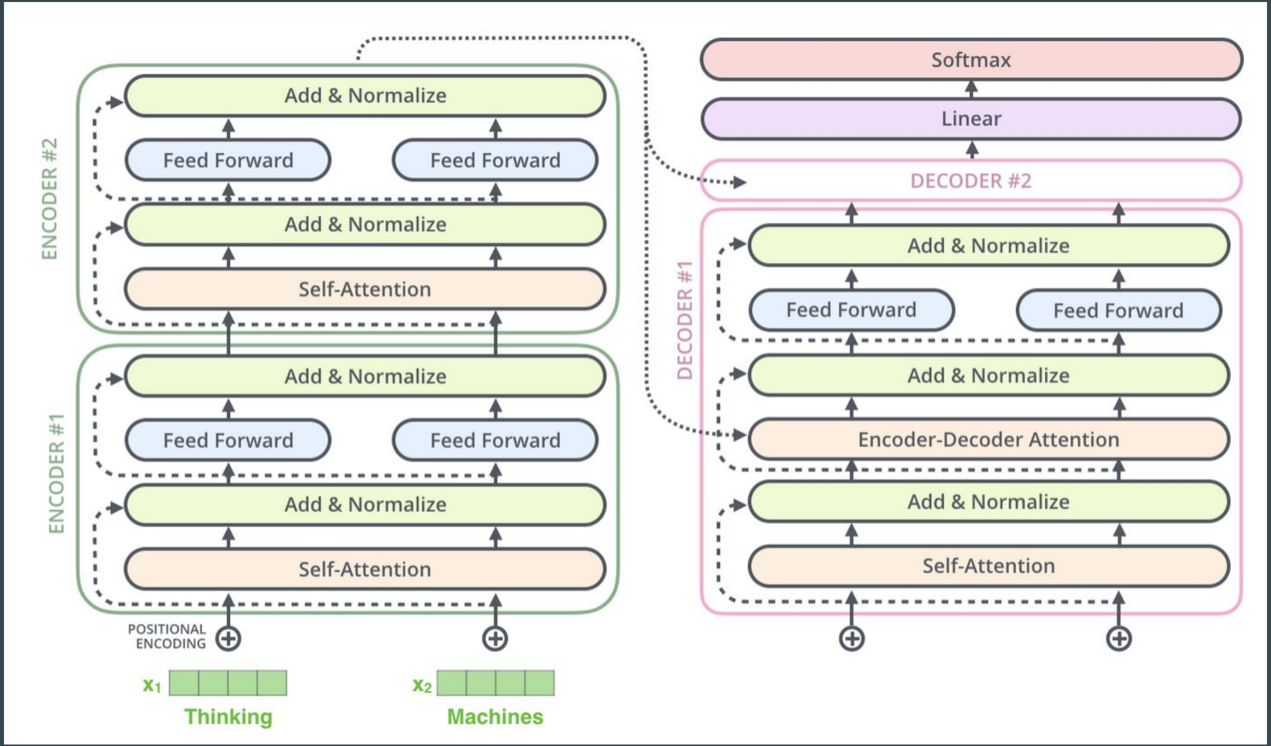
Idea: Do not hardcode context relationships through LSTMs, instead let the network learn which parts of the sentence are important in relation to each other (Attention)



Transformers - Implementation

- encoder - decoder network structure, no recurrence -> can be parallelized
- input is word embedding together with position signal
- encoder and decoders both have self-attention layers
- attention consists of query, key and value matrices (= weights), they are all learned through training
- multi-head attention is used: 8 independent attention layers per decoder block allows to focus on different parts of the sentence
- encoder-decoder attention allow decoder to access knowledge from encoder
- the whole network contains the information, not just the output layer

Transformers - Implementation



Transformers - Limitations

- not bidirectional, masks left context
- encoder-decoder structure only works for sequence to sequence tasks like translation or syntax parsing, unclear how to do classification tasks
- limitation of transfer learning due to task specific training

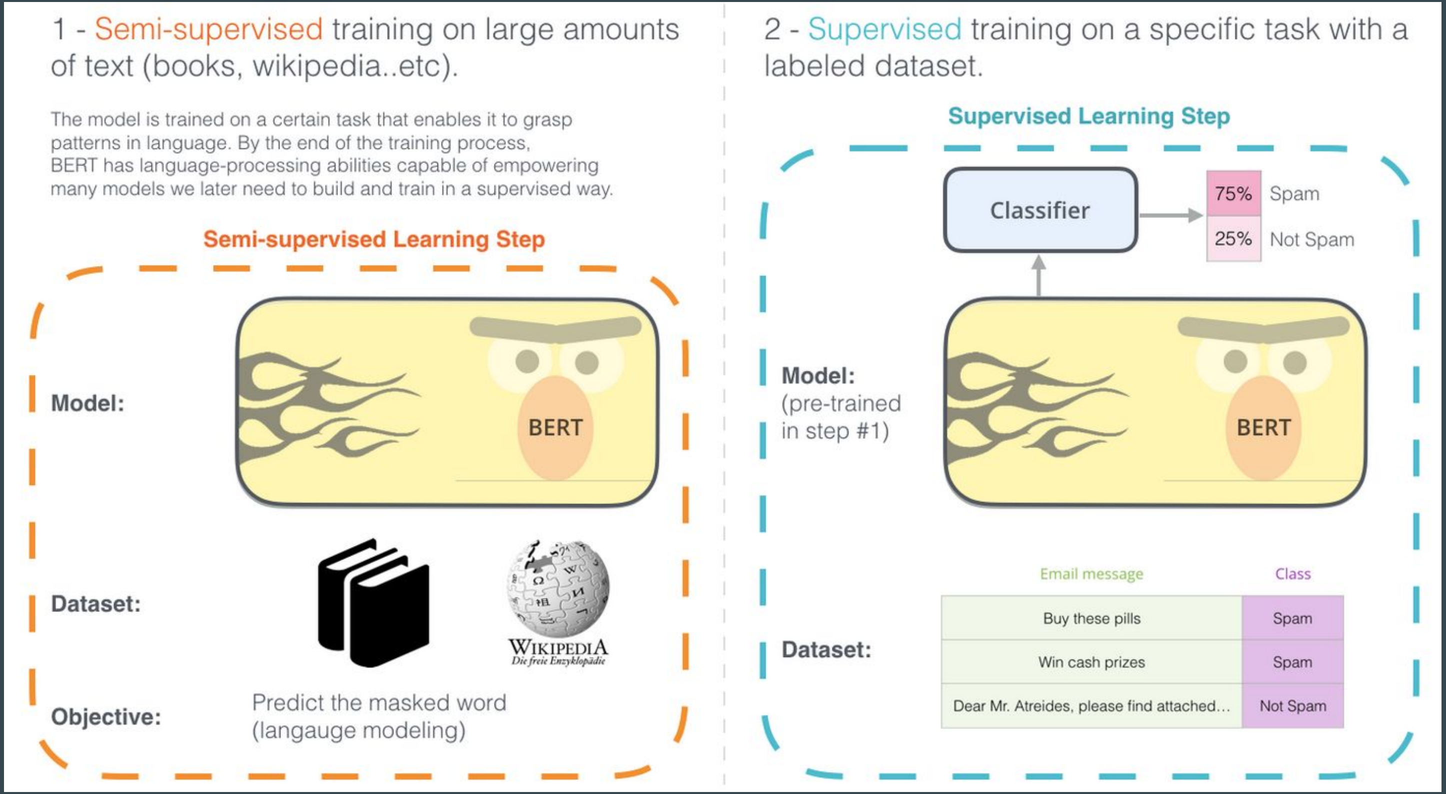
➡ some of these problems were fixed with the Open AI Transformer (GPT) (not in this presentation)

GPT

building on it: Open AI Transformers: using only the decoder, framework for pretraining and then fine tuning it for different tasks, all parameters are fine tuned

BERT

BERT - Intuition



1 - Semi-supervised training on large amounts of text (books, wikipedia..etc).

The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build and train in a supervised way.

Semi-supervised Learning Step

Model:

Dataset:

Objective: Predict the masked word (langauge modeling)

2 - Supervised training on a specific task with a labeled dataset.

Supervised Learning Step

Model: (pre-trained in step #1)

Dataset:

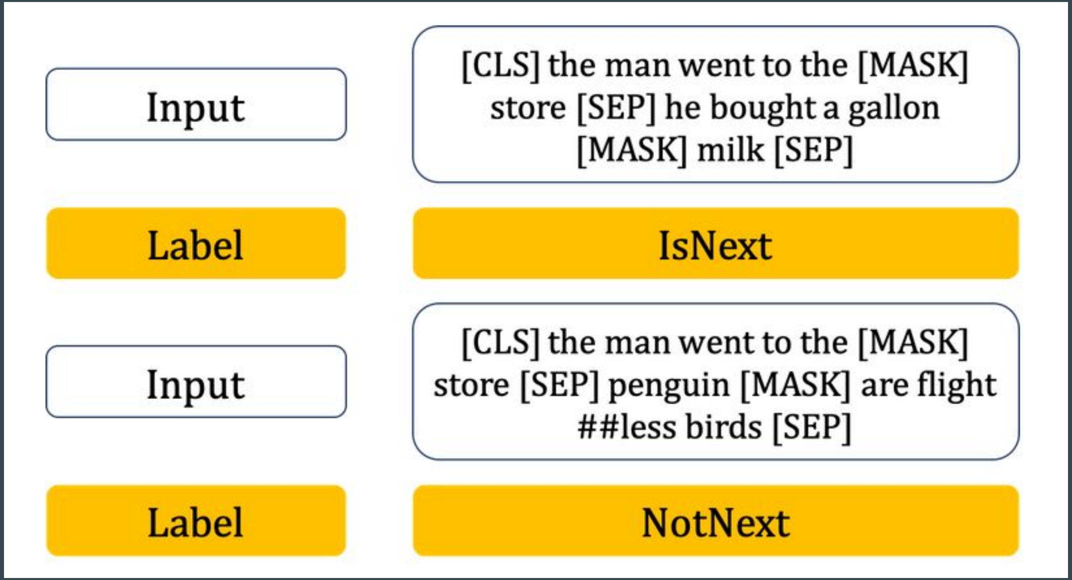
Email message	Class
Buy these pills	Spam
Win cash prizes	Spam
Dear Mr. Atreides, please find attached…	Not Spam

BERT - Implementation

- BERT stands for Bidirectional Encoder Representations from Transformers
  - (deeply) bidirectional because it takes full context into account
  - only using encoder blocks from Transformers → can be used for classification
- trained on 2 tasks:
  - masked language model: predict a masked token by its context
  - next sentence prediction: is the second sentence of a pair the successor of the first?
- successful transfer learning:
  - BERT is pretrained on huge unlabeled dataset (Books Corpus and English Wikipedia)
  - for a specific task, output layer is added and network can be fine-tuned
  - OR contextualized embeddings can be extracted and fed into other models
- Embeddings are distributed in whole model (in all 12 (Bert_base) or 24 (Bert_large) layers)

BERT - Implementation

Training data for BERT

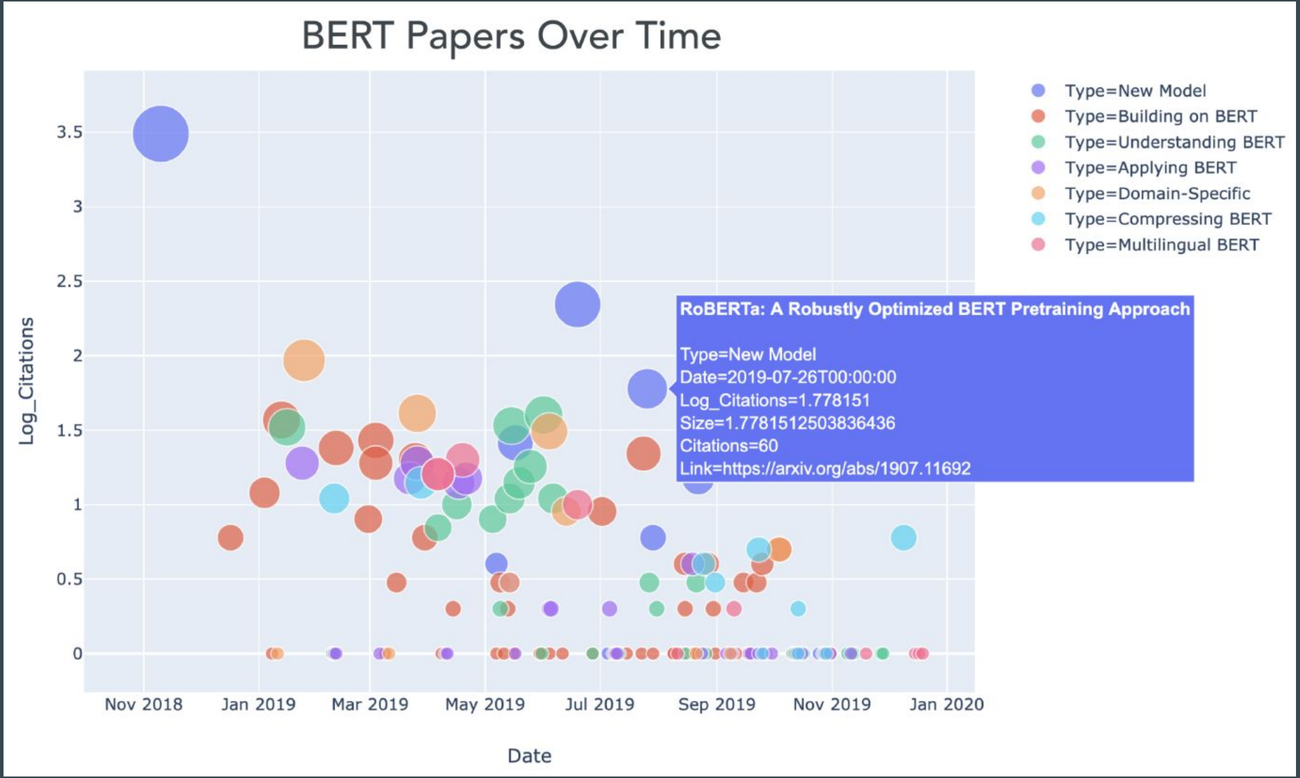


Input	[CLS] the man went to the [MASK] store [SEP] he bought a gallon [MASK] milk [SEP]
Label	IsNext
Input	[CLS] the man went to the [MASK] store [SEP] penguin [MASK] are flight ##less birds [SEP]
Label	NotNext

BERT - Use cases

- in general all kinds of classification tasks:
  - sentiment analysis
  - spam detection
  - fact checking
- question Answering tasks
- textual entailment
- semantic similarity search
- linguistic acceptability taks

BERT based Architectures



RoBERTa: A Robustly Optimized BERT Pretraining Approach

Type=New Model
Date=2019-07-26T00:00:00
Log_Citations=1.778151
Size=1.7781512503836436
Citations=60
Link=https://arxiv.org/abs/1907.11692

BERT based Architectures

RoBERTa: A Robustly Optimized BERT Pretraining Approach

- improves training procedure of BERT and uses different dataset for pretraining

DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter

- has 40% less parameters while still reaching 97% performance

ALBERT: A Lite BERT for Self-supervised Learning of Language Representations

- introduce parameter reducing techniques and new training task SOP (sentence order prediction)

How get started with BERT

- several pretrained networks available, eg. BERT-as-a-service, huggingface
- you can fine tune BERT for your dataset and add output layer for specific task
- other option: freeze weights, extract contextualized embeddings from one or more layers and use them as input for task specific network
    - here it makes sense to try different combinations like concatenation, average-pooling, max-pooling etc.
    - in general, earlier layers resemble more word embeddings, later layers are highly contextualized

Resources for using BERT

https://huggingface.co/transformers/index.html

https://github.com/hanxiao/bert-as-service

https://mccormickml.com/2019/07/22/BERT-fine-tuning/

https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/

Blog Articles

https://jalammar.github.io/illustrated-transformer/

http://jalammar.github.io/illustrated-bert/

https://towardsdatascience.com/2019-the-year-of-bert-354e8106f7ba

https://towardsdatascience.com/from-pre-trained-word-embeddings-to-pre-trained-language-models-focus-on-bert-343815627598

https://medium.com/analytics-vidhya/implementing-word2vec-in-tensorflow-44f93cf2665f

https://towardsdatascience.com/intuitive-explanation-of-bert-bidirectional-transformers-for-nlp-cdc1efc69c1e

https://machinelearnit.com/2019/08/19/bert-bidirectional-transformers-for-language-understanding/

https://humboldt-wi.github.io/blog/research/information_systems_1920/bert_blog_post/

Papers

Mikolov, Tomas; Sutskever, Ilya; Chen, Kai; Corrado, Greg; Dean, Jeffrey (2013): Distributed Representations of Words and Phrases and their Compositionality. http://arxiv.org/pdf/1310.4546v1

Peters, Matthew E.; Neumann, Mark; Iyyer, Mohit; Gardner, Matt; Clark, Christopher; Lee, Kenton; Zettlemoyer, Luke (2018): Deep contextualized word representations. https://arxiv.org/pdf/1802.05365

Vaswani, Ashish; Shazeer, Noam; Parmar, Niki; Uszkoreit, Jakob; Jones, Llion; Gomez, Aidan N. et al. (2017): Attention Is All You Need. http://arxiv.org/pdf/1706.03762v5

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, OpenAI. https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

Jacob Devlin; Ming-Wei Chang; Kenton Lee; Kristina Toutanova (2019): BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://arxiv.org/pdf/1810.04805.pdf

Papers

Liu, Yinhan; Ott, Myle; Goyal, Naman; Du, Jingfei; Joshi, Mandar; Chen, Danqi et al. (2019): RoBERTa: A Robustly Optimized BERT Pretraining Approach. https://arxiv.org/pdf/1907.11692.

Sanh, Victor; Debut, Lysandre; Chaumond, Julien; Wolf, Thomas (2019): DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. https://arxiv.org/pdf/1910.01108.

Lan, Zhenzhong; Chen, Mingda; Goodman, Sebastian; Gimpel, Kevin; Sharma, Piyush; Soricut, Radu (2019, October 30): ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. https://arxiv.org/pdf/1909.11942v6.pdf

Thank you for your Attention :)