Embedding, BERT and Applications in EHR

Alex Wang
Alex.w@wustl.edu
2021.05.21

- Applications
- Data Sets
- Embedding Models
- BERT
- Pre-Trained Model

Applications

- Information Extraction
- Outcome Prediction
- Relation Extraction
- Representation Learning

Information Extraction

- **Single Concept Extraction:** Sequence Labeling Task, e.g.: lab test, treatment, Data De-Identification
- **Temporal Event Extraction:** Assigning notions of time
- **Text Summary:** e.g.: Understanding patient complaint
- **Phenotyping:** e.g.: the broad characterization of patients' conditions
- **Abbreviation Expansion & Disambiguation**

Outcome Prediction

- e.g.: Disease diagnosis prediction (Most Common), mortality prediction

Relation Extraction

- **Clinical Relation Extraction:** Syntactic analysis

Representation Learning

- **Concept Representation:** Capturing the latent similarities between medical concepts
- **Patient Representation:** Obtaining vector representations of patients, improve model generality

Data Sets

EHR/EMR Data Sets

- **MIMIC**: Medical Information Mart for Intensive Care, By MIT Laboratory for Computational Physiology and collaborating research
- **I2b2**: Clinic Notes, By National NLP Clinical Challenges by Department of Biomedical Informatics, Harvard Medical School
- **KHRESMOI Project**: Biomedical texts and images, Founded by European Union, a multilingual, multimodal search and access system for biomedical information and documents
- **MADE**: NLP Challenges for Detecting **M**edication and **A**dverse **D**rug **E**vents, By University of Massachusetts
- **Sem**antic **Eval**uation, **T**emporal **H**istories of **Y**our **M**edical **E**vent, **P**arkinson **P**rogression **M**arker **I**nitiative, **A**lzheimer’s **D**isease **N**euroimaging **I**nitiative, eICU Collaborative Database, MEDLINE, …

Other Text Data Sets

- SHARP Seed [Seed] and SHARP Stratified [Strat]
- Create From Quro bot, an AI-driven clinical conversational platform
- NCBI Disease Corpus
- BioCreative-V-CDR-Corpus
- China Conference on Knowledge Graph and Semantic Computing EMRs corpus
- MiPACQ: Multi-source Integrated Platform for Answering Clinical Questions corpus
- TAC2017ADR: Adverse Drug Reaction Extraction from Drug Labels, By FDA.

Embedding Models

- What is Embedding and Why Using It?
- Non-contextual Embeddings
- Contextual Embeddings

What is Embedding?

Using Numbers to Represent Texts

Number: Number, vectors or matrix

Texts: Characters, words or sentences

Why Using Embedding?

Deep Learning Models Don’t Understand Text
They Only Take Number as Input

Why Using Embedding?

Embeddings can modeling lingustic features
and help the model to learn further information

Distributed Representation (Neuron Network)

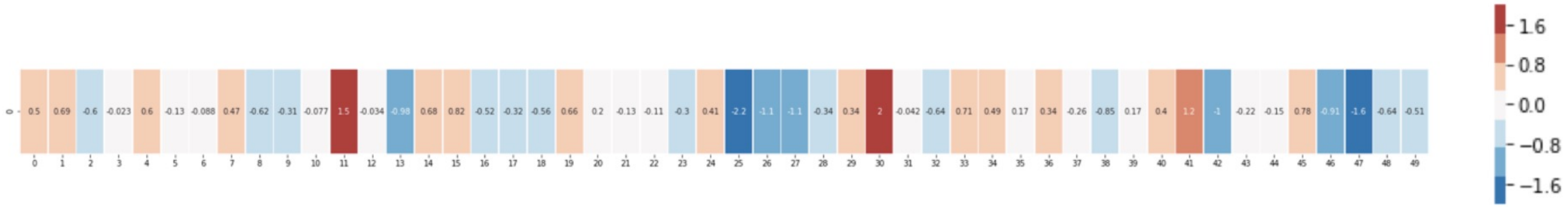
- Describe the meaning of a piece of text by low-dimensional real-valued vectors.
- Models: Word2Vec, Sentence2Vec, Context2Vec

Distributional Representation (Statistics)

- Describe the meaning of a piece of text in matrix space
- Model: GloVe

Word2Vec

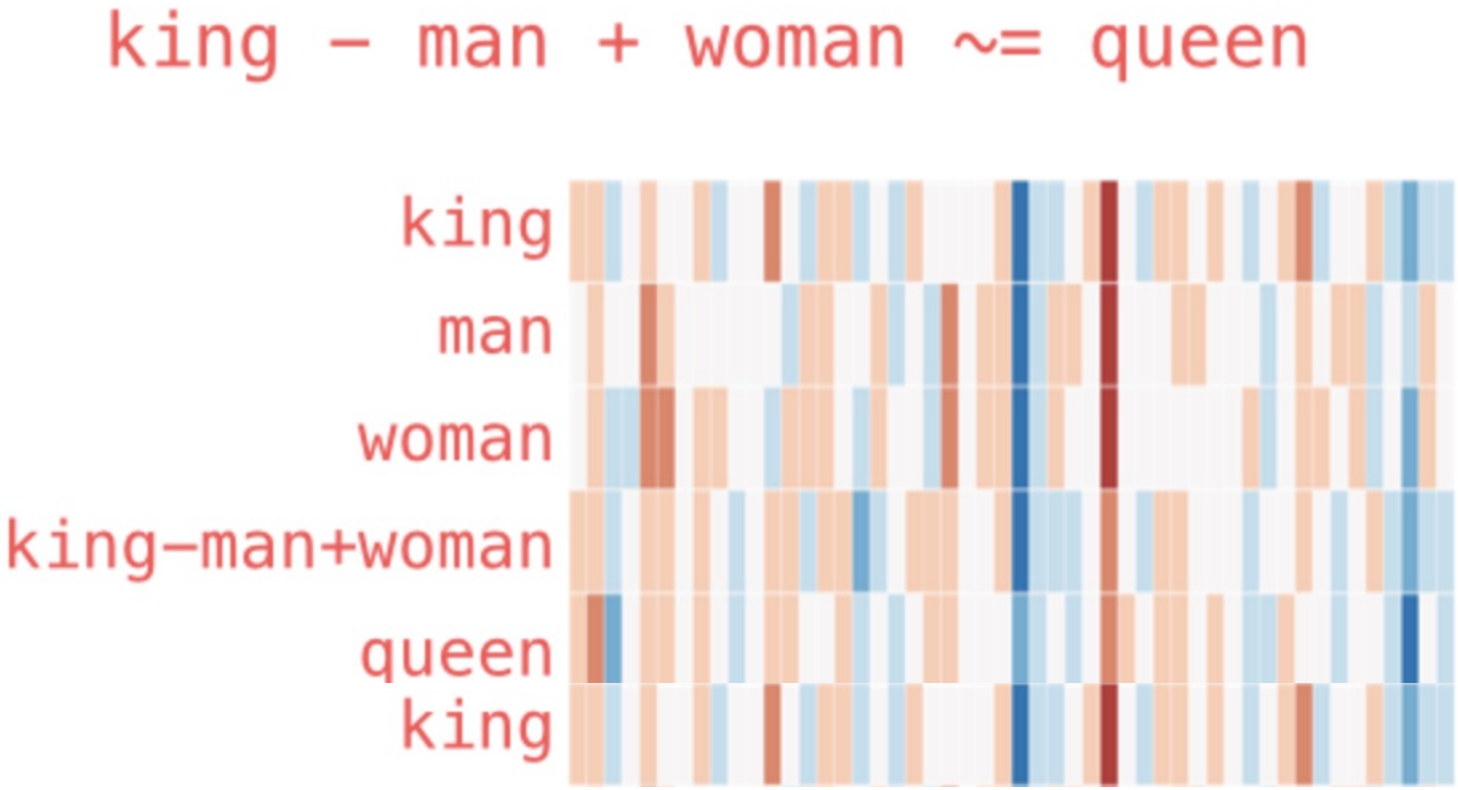
Word2Vec Result for word King



Picture From: https://jalammar.github.io/illustrated-word2vec/

Non-contextual Embeddings

Word2Vec



Picture From: https://jalammar.github.io/illustrated-word2vec/

Word2Vec

```
model.most_similar(positive=["king","woman"], negative=["man"])
```

```
[('queen', 0.8523603677749634),
 ('throne', 0.7664333581924438),
 ('prince', 0.7592144012451172),
 ('daughter', 0.7473883032798767),
 ('elizabeth', 0.7460219860076904),
 ('princess', 0.7424570322036743),
 ('kingdom', 0.7337411642074585),
 ('monarch', 0.721449077129364),
 ('eldest', 0.7184862494468689),
 ('widow', 0.7099430561065674)]
```

Pros (Word2Vec)

- Can capture both syntactic and semantic word relationships

$$\text{vec}(\text{“China”}) - \text{vec}(\text{“Beijing”}) \approx \text{vec}(\text{“Japan”}) - \text{vec}(\text{“Tokyo”})$$

- No need for **deep** neural networks to build good word embeddings
- Word2vec embeddings implicitly encode referential attributes of entities

Cons

- Static embedding, not based on context (Words with different meaning)
- Fail to capture syntactic structures and other linguistic features.

Pros (Word2Vec)

- Can capture both syntactic and semantic word relationships
- No need for deep neural networks to build good word embeddings
- Word2vec embeddings implicitly encode referential attributes of entities

Referential Attributes:
The population, GDP of Italy are
referential attributes for "Italy"

Cons

- Static embedding, not based on context (Words with different meaning)
- Fail to capture syntactic structures and other linguistic features.

Sequence Models

- Convolutional Models: Aggregating the local information from its neighbors by convolution operations
- Recurrent Models: LSTMs and GRUs

Pros & Cons

- Pros: Easy to train and get good results for various NLP tasks
- Cons: Based on neighbor words or suffered from long-term dependency

Non-Sequence Models

- Fully-Connected Self-Attention Model: Transformer
- Predefined Structure: Recursive NN, TreeLSTM, and Graph Convolutional Network (GCN)

Pros & Cons

- Pros: Suitable to model long range dependency of language
- Cons:
  - Transformer is easy to overfit
  - Good predefined structure is hard to find
  - Taking too long to train

BERT

- **B**idirectional **E**ncoder **R**epresentation from **T**ransformers

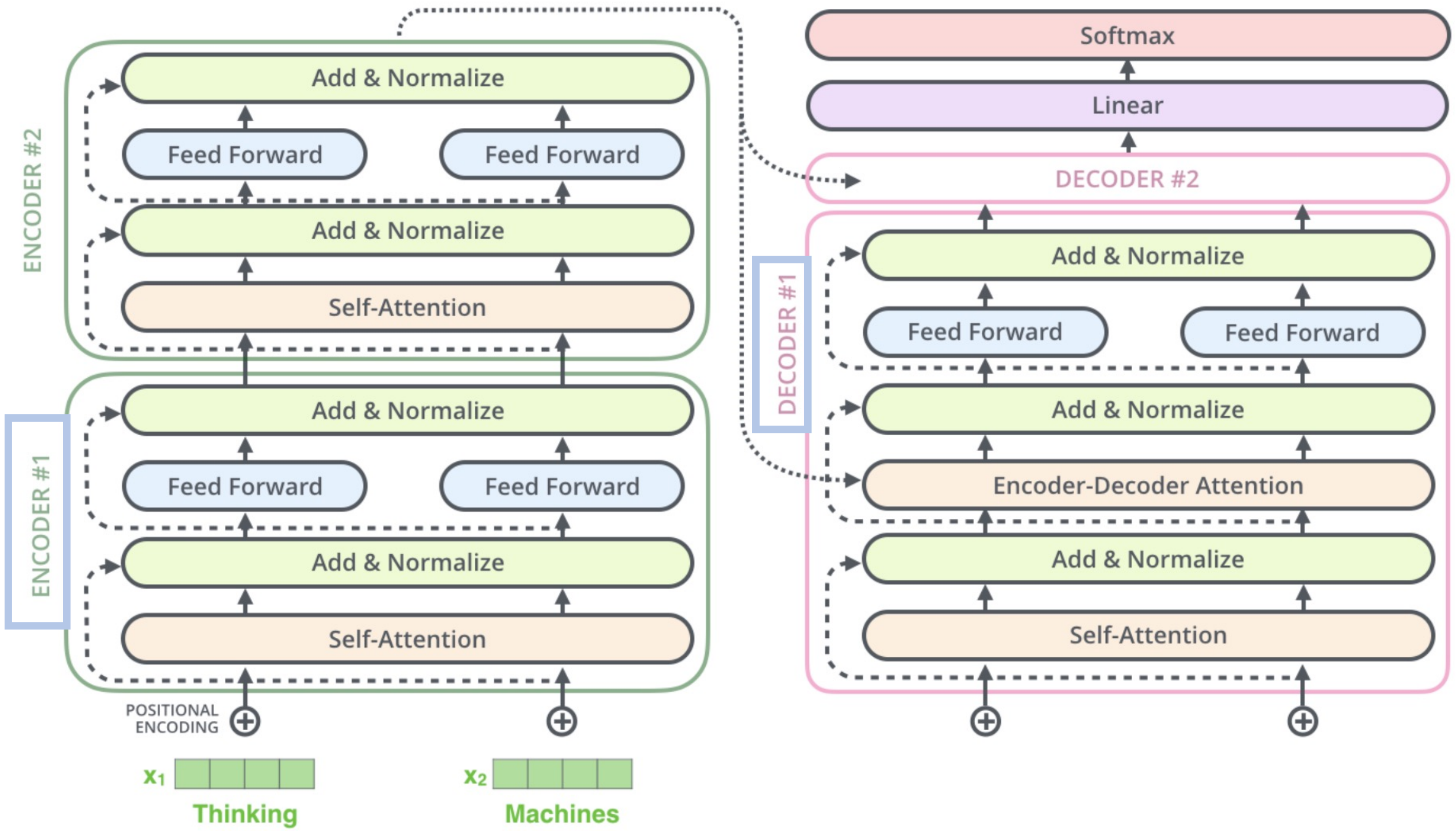
Pros

- Good performance on syntactic tasks
- Ability to learn subject-verb agreement and semantic roles (Compare to other models)
- Ability to encode syntax structure

Cons

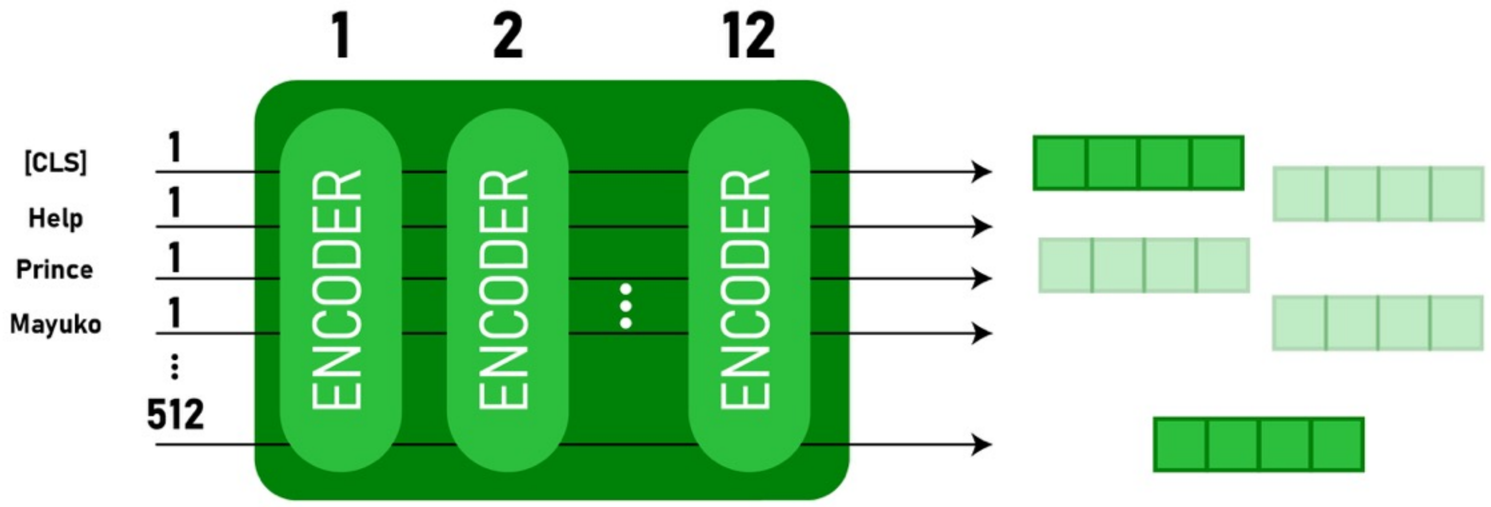
- Not good enough at semantic and fine-grained syntactic tasks
- Good at understanding but not so good at generating

Transformer



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

BERT

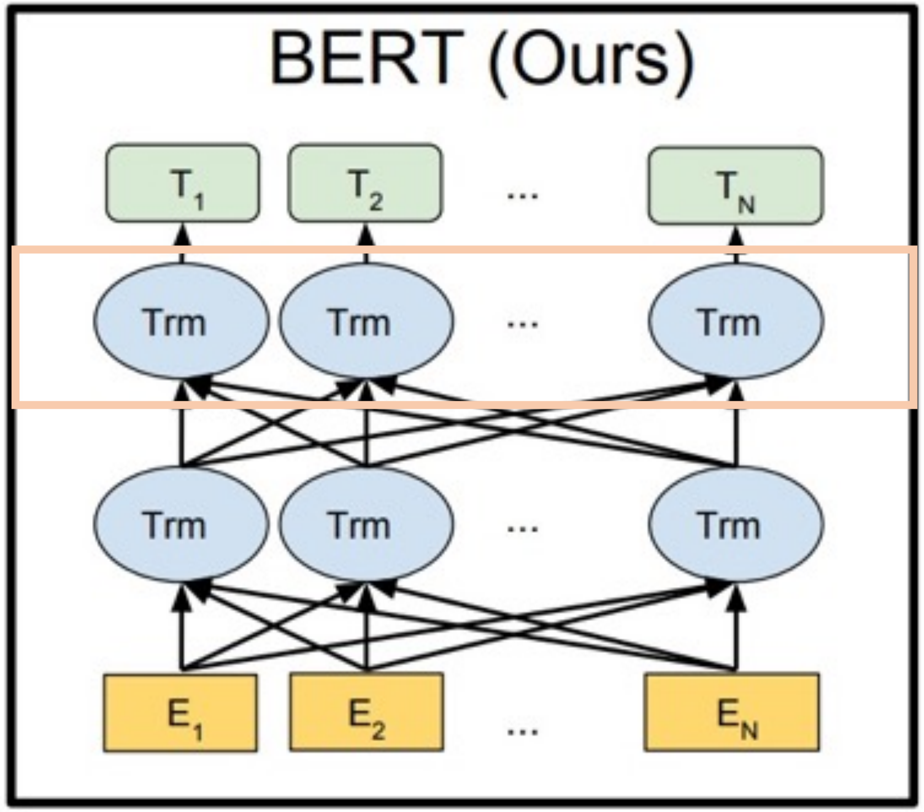


BERT output as Embeddings

Picture From: https://www.geeksforgeeks.org/explanation-of-bert-model-nlp/

BERT

Transformer
Encoder



Picture From: https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework/

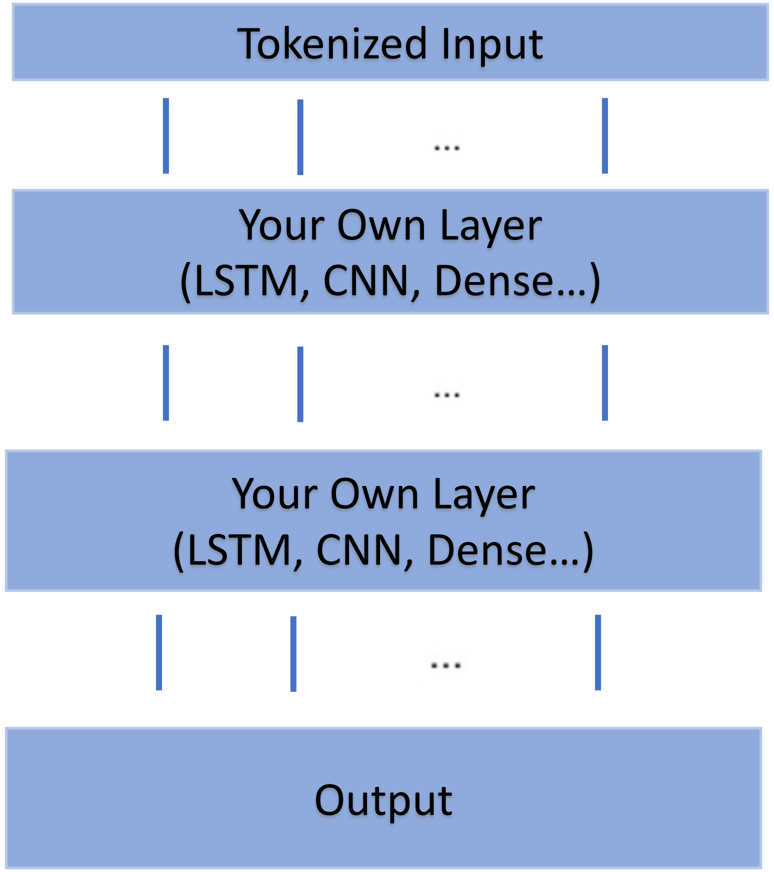
Pre-Training

- Why Pre-training?
- How to use Pre-Trained model?

Why Pre-training

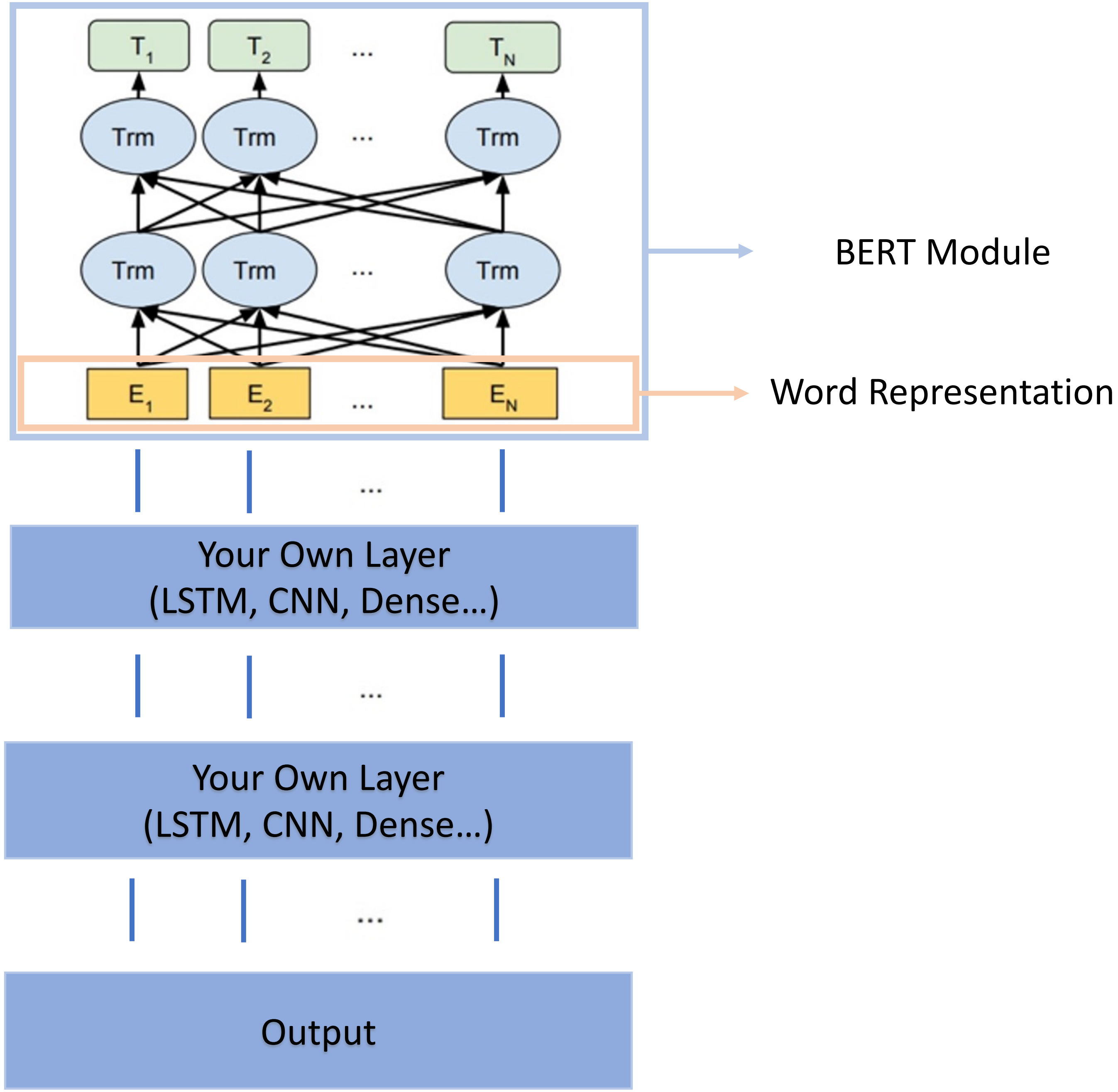
- Provides better model initialization
- Speeds up convergence on the target task
- Can be regarded as a regularization method and avoid overfitting

How to use
Pre-Trained model



Pre-training

How to use
Pre-Trained model



How to use Pre-Trained model

```python
model_name='cambridgeltl/BioRedditBERT-uncased'

def tokenize_data(data,model_name):
    tokenizer = BertTokenizer.from_pretrained(model_name)
    encoded = tokenizer.batch_encode_plus(
        data,
        add_special_tokens=True,
        max_length=max_length,
        return_attention_mask=True,
        return_token_type_ids=True,
        pad_to_max_length=True,
        return_tensors="tf",
    )

    return np.array(encoded["input_ids"], dtype="int32")
```

How to use Pre-Trained model

```
max_length=128
num_tags=10
```

Here we are doing a sequence labeling task

Input Layers

```
input_ids = tf.keras.layers.Input(shape=(max_length), dtype=tf.int32, name="input_ids")
```

BERT

```
embedding = transformers.TFBertModel.from_pretrained(model_name)
embedding.trainable = False     # we dont want to train embedding
out = embedding(input_ids)
embedding_output = out[0]
```

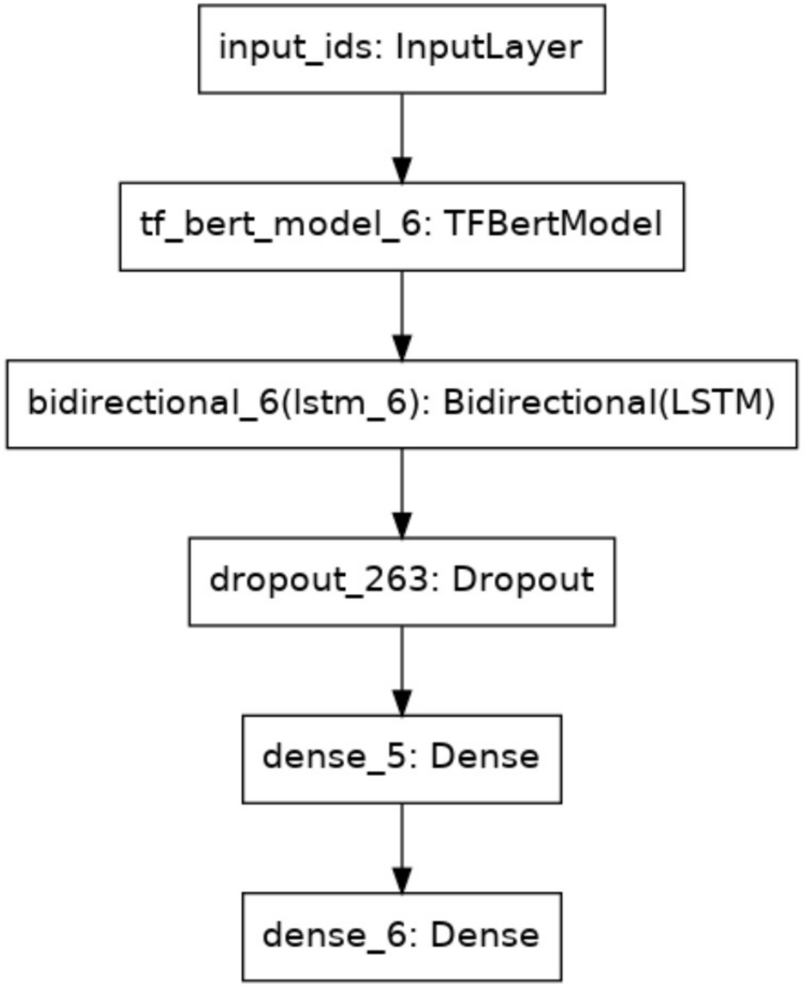
Your Layers

```
bi_lstm = tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(64, return_sequences=True))(embedding_output)
dropout = tf.keras.layers.Dropout(0.3)(bi_lstm)
dense = tf.keras.layers.Dense(32)(dropout)
output = tf.keras.layers.Dense(num_tags, activation="softmax")(dense)
```

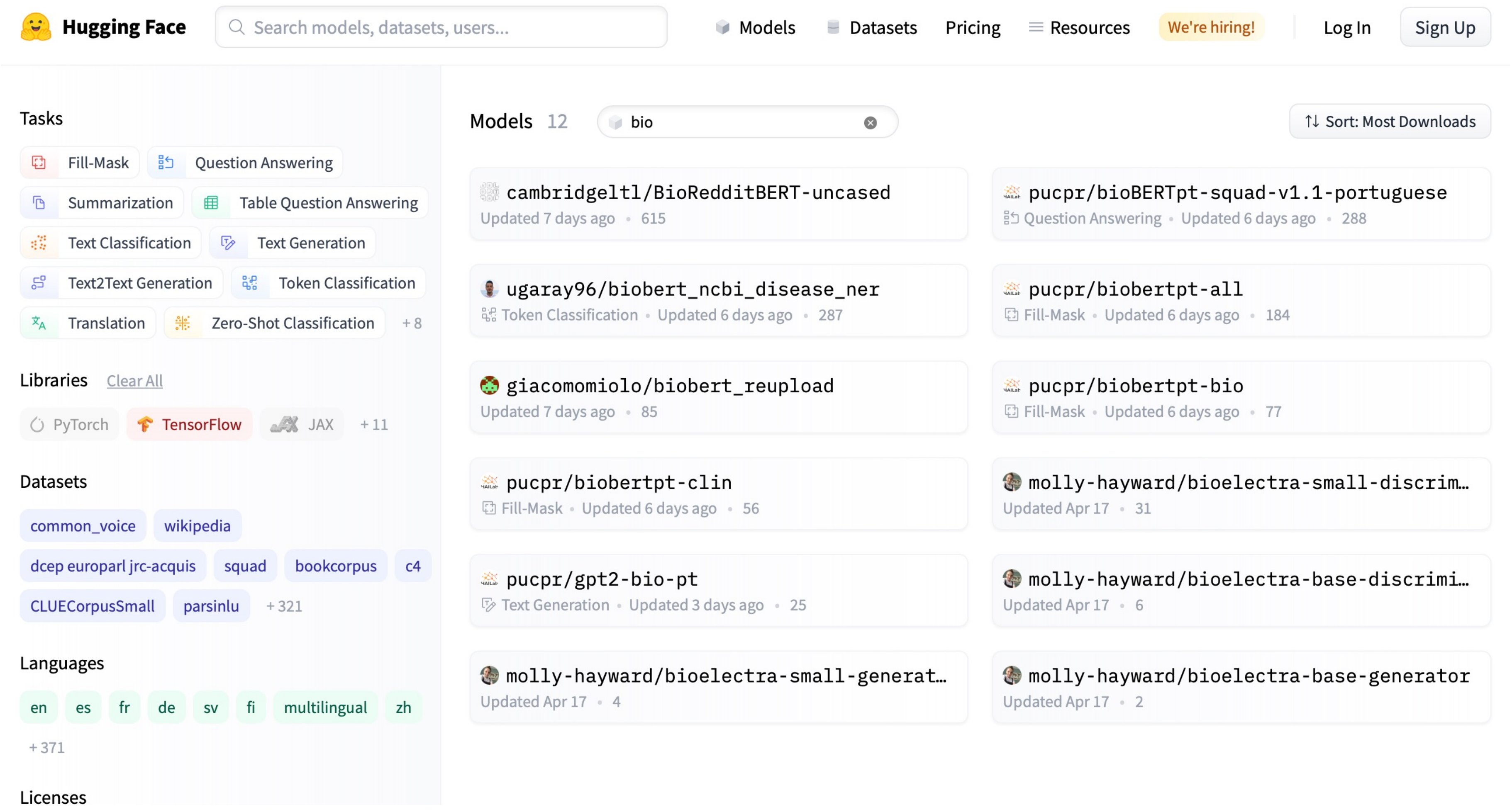
```
model = tf.keras.models.Model(inputs=[input_ids], outputs=output)
model.compile(optimizer=tf.keras.optimizers.Adam(),loss="categorical_crossentropy",metrics=["acc"])
tf.keras.utils.plot_model(model)
```

```
Downloading: 100% 462/462 [00:00<00:00, 14.3kB/s]
Downloading: 100% 434M/434M [00:19<00:00, 23.2MB/s]
```

How to use Pre-Trained model



Pre-training



https://huggingface.co/models?filter=tf&search=bio

Questions?

Thanks

Reference

- Esteva, A., Robicquet, A., Ramsundar, B. et al. A guide to deep learning in healthcare. Nat Med 25, 24–29 (2019). https://doi.org/10.1038/s41591-018-0316-z
- B. Shickel, P. J. Tighe, A. Bihorac and P. Rashidi, "Deep EHR: A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis," in IEEE Journal of Biomedical and Health Informatics, vol. 22, no. 5, pp. 1589-1604, Sept. 2018, doi: 10.1109/JBHI.2017.2767063.
- Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, Joel T Dudley, Deep learning for healthcare: review, opportunities and challenges, Briefings in Bioinformatics, Volume 19, Issue 6, November 2018, Pages 1236–1246, https://doi.org/10.1093/bib/bbx044
- Ji Z, Wei Q, Xu H. BERT-based Ranking for Biomedical Entity Normalization. AMIA Jt Summits Transl Sci Proc. 2020 May 30;2020:269-277. PMID: 32477646; PMCID: PMC7233044.
- Lin C, Bethard S, Dligach D, Sadeque F, Savova G, Miller TA. Does BERT need domain adaptation for clinical negation detection? J Am Med Inform Assoc. 2020 Apr 1;27(4):584-591. doi: 10.1093/jamia/ocaa001. PMID: 32044989; PMCID: PMC7075528.

Reference

- Kanerva J, Ginter F, Pyysalo S. Dependency parsing of biomedical text with BERT. BMC Bioinformatics. 2020 Dec 29;21(Suppl 23):580. doi: 10.1186/s12859-020-03905-8. PMID: 33372589; PMCID: PMC7771067.
- B. Saha, S. Lisboa and S. Ghosh, "Understanding patient complaint characteristics using contextual clinical BERT embeddings," 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), 2020, pp. 5847-5850, doi: 10.1109/EMBC44109.2020.9175577.
- Li F, Jin Y, Liu W, Rawat BPS, Cai P, Yu H Fine-Tuning Bidirectional Encoder Representations From Transformers (BERT)–Based Models on Large-Scale Electronic Health Record Notes: An Empirical Study, JMIR Med Inform 2019;7(3):e14830, doi: 10.2196/14830