Lec 15 - Add knowledge to Language Models

1. Recap: language models (LMs)

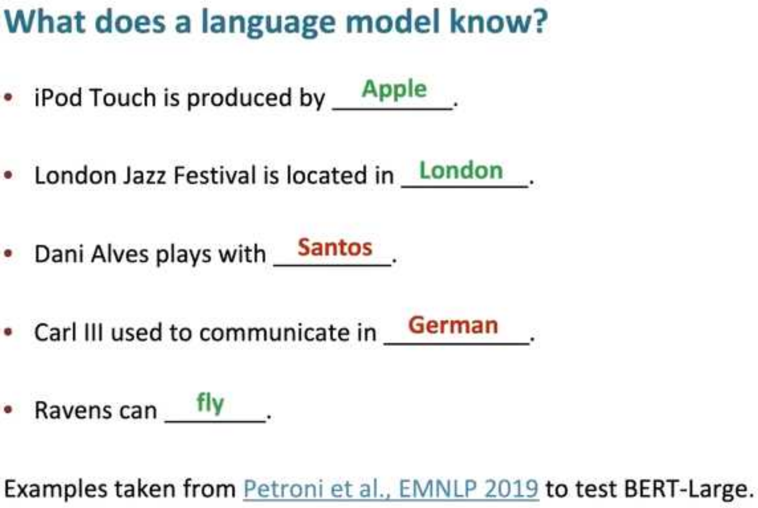
- Standard language models predict the next word in a sequence of text and can compute the probability of a sequence

- Recently, masked language models(e.g., BERT) instead predict a masked token in a sequence of text using bidirectional context

- Both type of language models can be trained over large amounts of unlabeled text!

- Traditionally, LMs are used for many tasks involving generating or evaluating the probability of text. Today, LMs are commonly used to generate pretrained representations of text that encode some notion of language understanding for downstream NLP tasks

2. What does a LM know?



What does a language model know?

- iPod Touch is produced by Apple.
- London Jazz Festival is located in London.
- Dani Alves plays with Santos.
- Carl III used to communicate in German.
- Ravens can fly.

Examples taken from Petroni et al., EMNLP 2019 to test BERT-Large.

predictions generally make sense but are not all factually correct.
Unseen facts, Rare facts, Model sensitivity, inability to reliably recall knowledge

- The importance of knowledge-aware language models
: benefit downstream tasks that leverage knowledge.
: replace traditional knowledge bases

- Querying traditional knowledge bases. (from WIKIDATA)

- pretrain LM over unstructured text and then query with NL.
- LMs are pretrained over large amounts of unstructured and unlabeled text
- LMs support more flextible natural language queries
- but.. hard to interpret, trust, modify

3. Techniques to add knowledge to LMs

-1. Add pretrained entity embeddings

: Facts about the world are usually in terms of entities

: pretrained word embeddings do not have a notion of entities

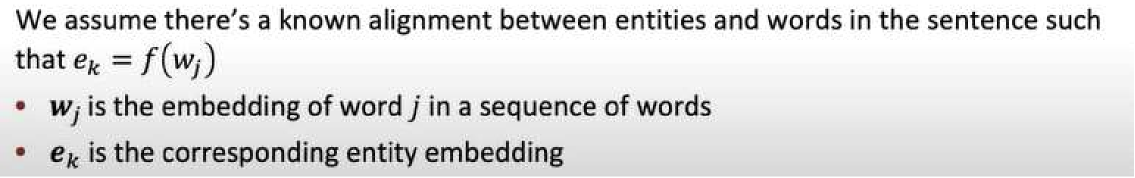
 -> Different word embeddings

 -> Single entity embedding

- What is entity linking? Link mentions in text to entities in a knowledge base
- which entity embeddings are relevant to the text (TransE, Wikipedia2Vec, BLINK)

-How do we incorporate pretrained entity embeddings from a different embedding space? -> fusion layer

$$\boldsymbol{h}_j = F\left(\boldsymbol{W}_t\boldsymbol{w}_j + \boldsymbol{W}_e\boldsymbol{e}_k + b\right)$$



We assume there's a known alignment between entities and words in the sentence such that $e_k = f(w_j)$

- $\boldsymbol{w}_j$ is the embedding of word j in a sequence of words
- $\boldsymbol{e}_k$ is the corresponding entity embedding

- ERNIE

: Text encoder: multi-layer bidirectional Transformer encoder over the words in the sentence

: Knowledge encoder: stacked blocks composed of MHAs, fusion layer

- KnowBERT

-2. Use an external memory

-3. Modify the training data

4. Evaluating knowledge in LMs