# COVID Literature Search

COMS W4995 Applied Deep Learning

Tim Huang (thh2114) and Jay Zern Ng (jn2717)

# Motivation

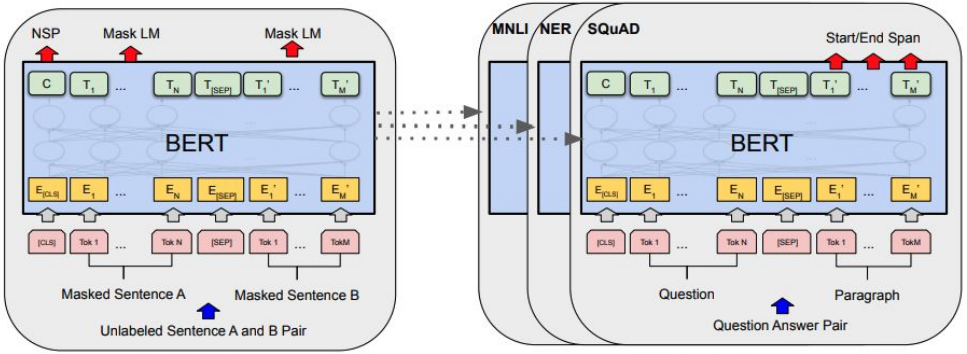
- **Information retrieval**: i.e. COVID-19 Open Research Dataset (CORD-19) by the Allen Institute for AI, over 29,000 scholarly articles and 13,000+ in full text. (https://allenai.org/)
- **Semi-supervised learning:** rising state of the art methods such as BERT, T5, SimCLR, OpenAI GPT-2 and more. (https://ai.googleblog.com/2020/04/advancing-self-supervised-and-semi.html)
- **Accessibility**: Bridging the gap between deep learning practitioners and medical researchers.

# Contributions

- Application:
  - Built a search engine for COVID-19 research journals + scoring system for **results**.
  - Rolling **suggestions** based on search **context** to prompt users.
  - Visualization tool to interpret **keywords** using BERT.
- Tools Used:
  - PyTorch (HuggingFace)
  - React.js + Flask
  - Google Cloud Platform

# Key Concepts I

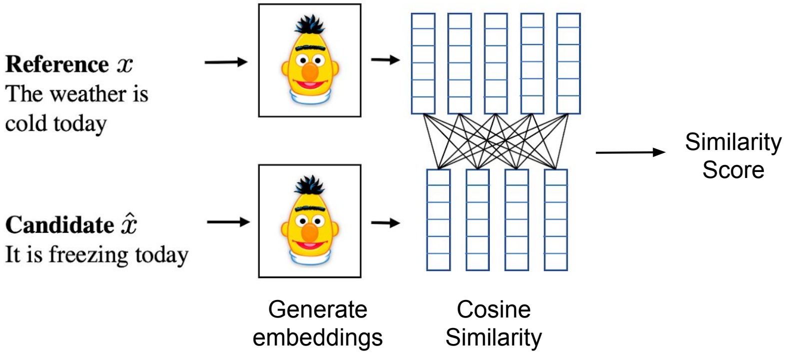
- BERT (Devlin, et al., 2018)
  - **Pre-training:** BooksCorpus (800M words) and Wikipedia (2,500M words).
  - **Embeddings:** Tokens ([CLS] at the start, [SEP] at the end), Segments (A or B) and Positional.
  - **Masked Language Modeling (MLM):** randomly mask 15% of words with [MASK].
  - **Next Sentence Prediction (NSP):** separate sentences using [SEP], predict if two sentences are connected.
  - **Encoder and Decoder:** generate and retrieve encoded representations.




Pre-training Fine-Tuning

# Key Concepts II

- Sentence Embeddings
    - Only use **encoder** portion of BERT to extract **embedded** versions of sentences
    - Use **similarity function** to identify similar sentence embedding vectors
    - Normal BERT does not produce great embeddings for similarity comparison
    - **Fine-tune** using custom loss function based on similarity scores



# Sentence Embeddings

- CovidBERT:
  - Pre-trained model hosted on HuggingFace
  - BERT trained on same COVID-19 research dataset
  - Fine-tuned on Stanford NLI similarity dataset
- Pre-generated embeddings for entire Allen COVID-19 dataset
- Same model used to encode query into embeddings.
- Cosine similarity search

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|}$$

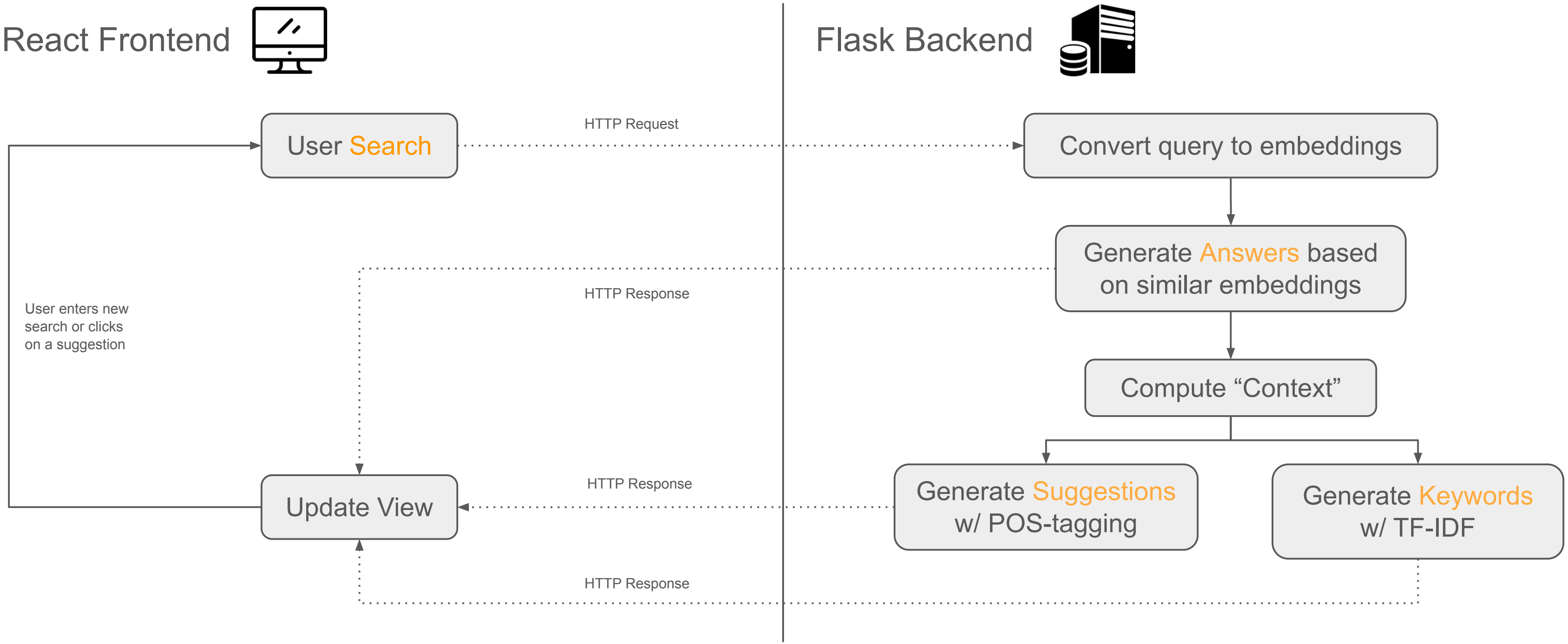
# Experiments

| Model | Parameters | Embedding Generation Time |
|---|---|---|
| BERT-base (Devlin, et al., 2018) | 110 million | 4 hours |
| RoBERTa (Liu et al., 2019) | 125 million | 4 hours |
| DistilBERT (Sanh et al., 2020) | 66 million | 2 hours |
| **CovidBERT [1][2]** | **110 million** | **3 hours** |

[1]: https://huggingface.co/gsarti/covidbert-nli
[2]: https://github.com/gsarti/covid-papers-browser

# Demo

# Architectural Design



# Summary

- What worked well:
  - Search works well for specific queries, but worse for general queries
  - Relatively fast, as long as there is sufficient memory
- Difficulties:
  - Memory limitation issues
  - Deployment difficulties
- Future directions:
  - Longer training time may lead to improvements
  - Approximate similarity search
  - Model Quantization
  - Deployment

# References

- Papers:
  
  - J. Devlin, M. Chang, K. Lee, K. Toutanova (2018), BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
  - A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin (2017), Attention Is All You Need
  - N. Reimers, I. Gurevych (2020), Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks
  
- Code:
  
  - https://huggingface.co/gsarti/covidbert-nli
  - https://github.com/UKPLab/sentence-transformers
  - https://stevenloria.com/tf-idf/
  - https://github.com/indrajithi/genquest
  

# Appendix I

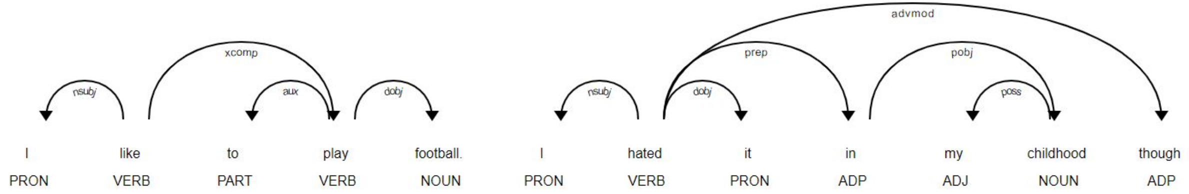
- TF-IDF
  - Bag of Words model that evaluates how important a word is to a document within a collection.
  - **Term Frequency:** how frequent a terms occurs in a document.

$$TF(t) = (\text{\#term t appears in document})/(\text{\#terms in a document})$$

  - **Inverse Document Frequency:** used to weigh down frequent terms and scale rare ones.

$$IDF(t) = \log[(\text{\#documents})/(\text{\#documents with term t})]$$

- Part-of-Speech Tagging
  - Generate suggestions by combining POS tags into a sensible parse tree.
  - Example: “What is COVID-19?” is the sequence [PRONOUN], [VERB], [NOUN], [?].



# Appendix II

- Transformer (Vaswani, et al., 2017)
  - **Key, Value and Query**: scaled dot-product attention.
  - **Multi-headed Self-Attention**: "jointly attend to information from different representation subspaces at different positions"
  - **Encoder and Decoder**: generate and retrieve encoded representations.

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{n}})\mathbf{V}$$

$$MultiHead(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [head_1; \ldots; head_h]\mathbf{W}^O$$

$$\text{where } head_i = Attention(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V)$$