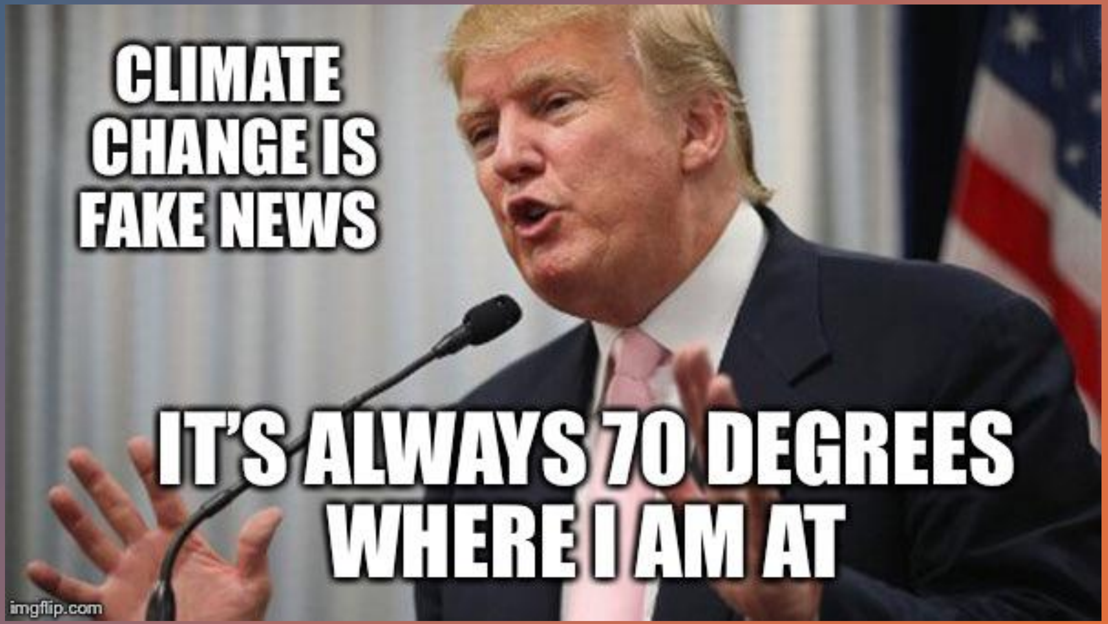
COMP 90042 Project

Automated Fact-Checking for Climate Claims

Mon5PM_Group2



Background & Motivation

- Climate change is a contentious subject.
- Misinformation is rampant, especially on social media and the internet in general.
- Stopping the spread of climate science misinformation is crucial to ensure informed public decision-making and effective climate action.
- Manual fact-checking by human experts infeasible as new misinformation constantly pouring in.
- Automation necessary.

Automation of Fact-Checking

Recent advances in machine learning (LLMs) have made it possible to automate **natural language understanding** and **knowledge-intensive** tasks.

Lot's of research in recent years on Open-Domain Question Answering and Automated Fact-Checking.

Reader-Retrieval Models for Factoid Question Answering

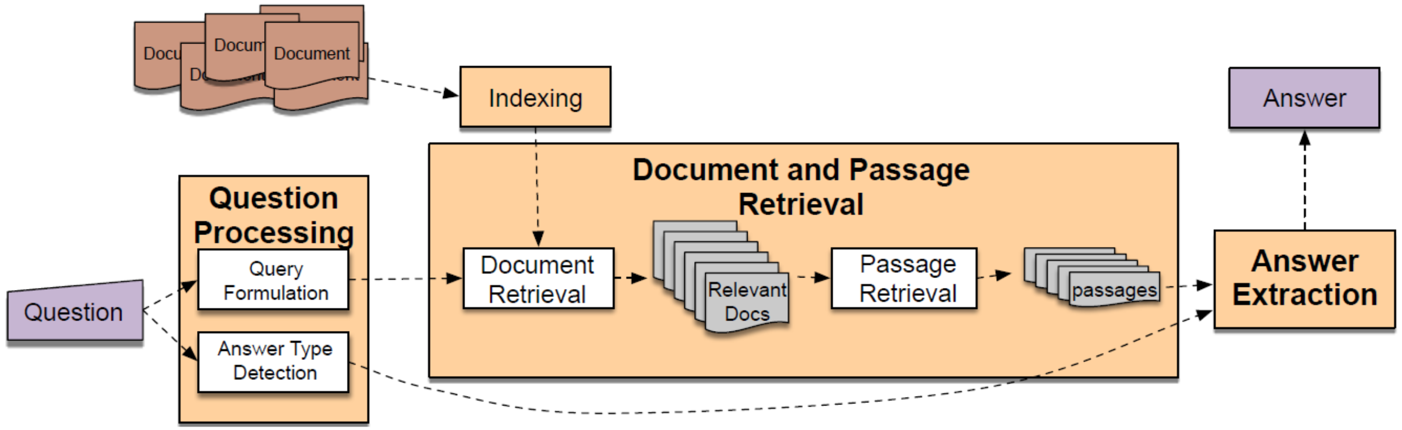


Image source: Jurafsky and Martin 2019, fig. 25.2

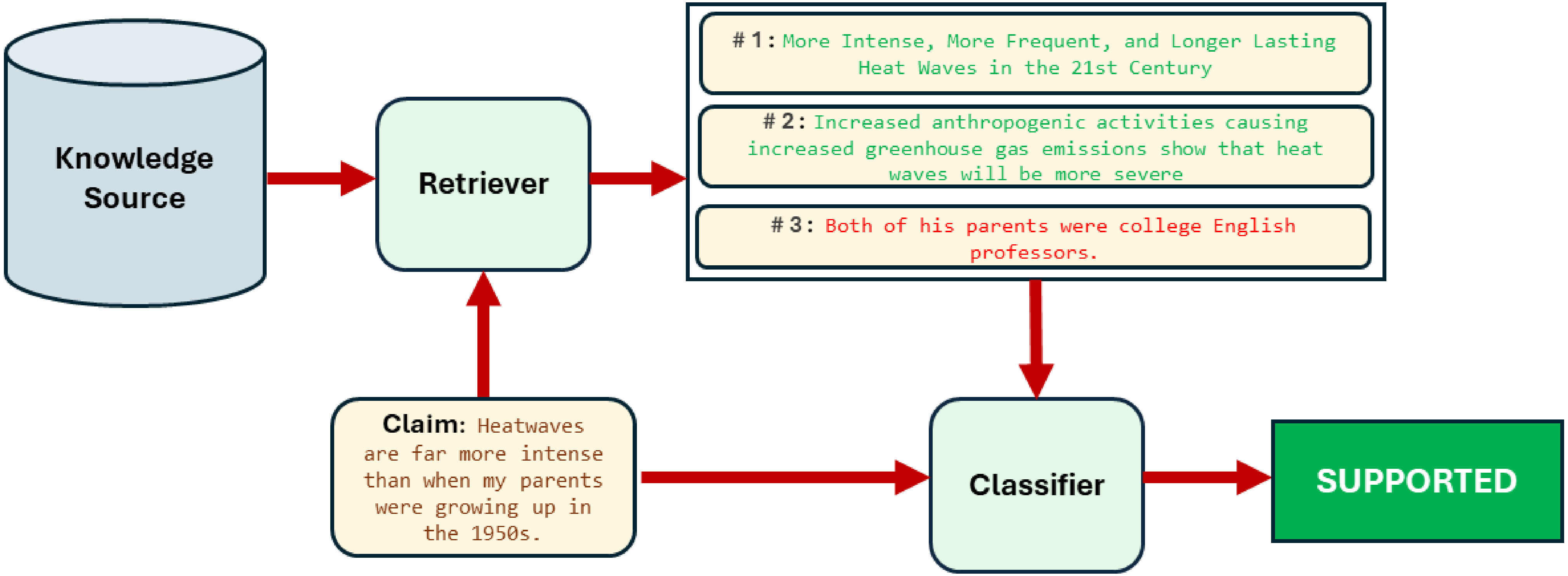
Project Main Goals

1) DESIGN AND IMPLEMENT AN AUTOMATED FACT VERIFICATION SYSTEM FOR CLIMATE RELATED CLAIMS USING STATE-OF-THE-ART MACHINE LEARNING TECHNIQUES

2) ACHIEVE RELIABLE PERFORMANCE, HIGH ACCURACY AND EFFICIENCY

Method

System Pipeline: Two Stage Approach



Stage 1) Retriever - Fetches relevant documents from knowledge source

Stage 2) Classifier - Classifies claim into one of the following : [SUPPORTS, REFUTES, NOT_ENOUGH_INFO, DISPUTED]

Retriever: BM25 with BERT Re-ranking

- BM25 is a variant of TFIDF
- Documents and query represented by sparse bag-of-words vectors
- Documents are ranked by dot product similarity score.
- Training corpus preprocessed and normalized: lower-case folding, stemming, stop words removed
- Very efficient, however scores rely on direct word matching, not always aligned with semantic content.

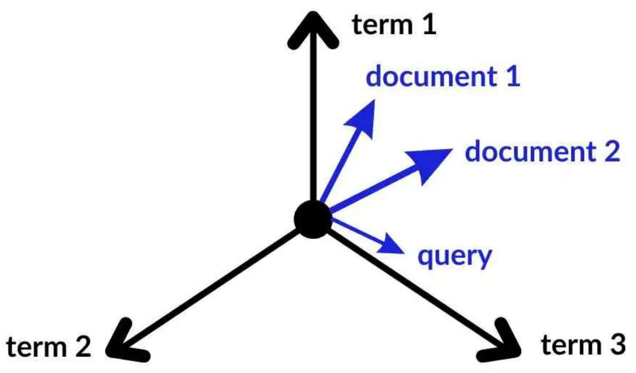


Image source: spotintelligence.com/2023/09/07/vector-space-model/

Query: **"Greenland enters melt mode"**	**Normalized Score**
Doc 1: "A sudden lurch into melting, Greenland's ice is shrinking"	0.75
Doc 2: "Greenland's ice is on the hot seat again"	0.3

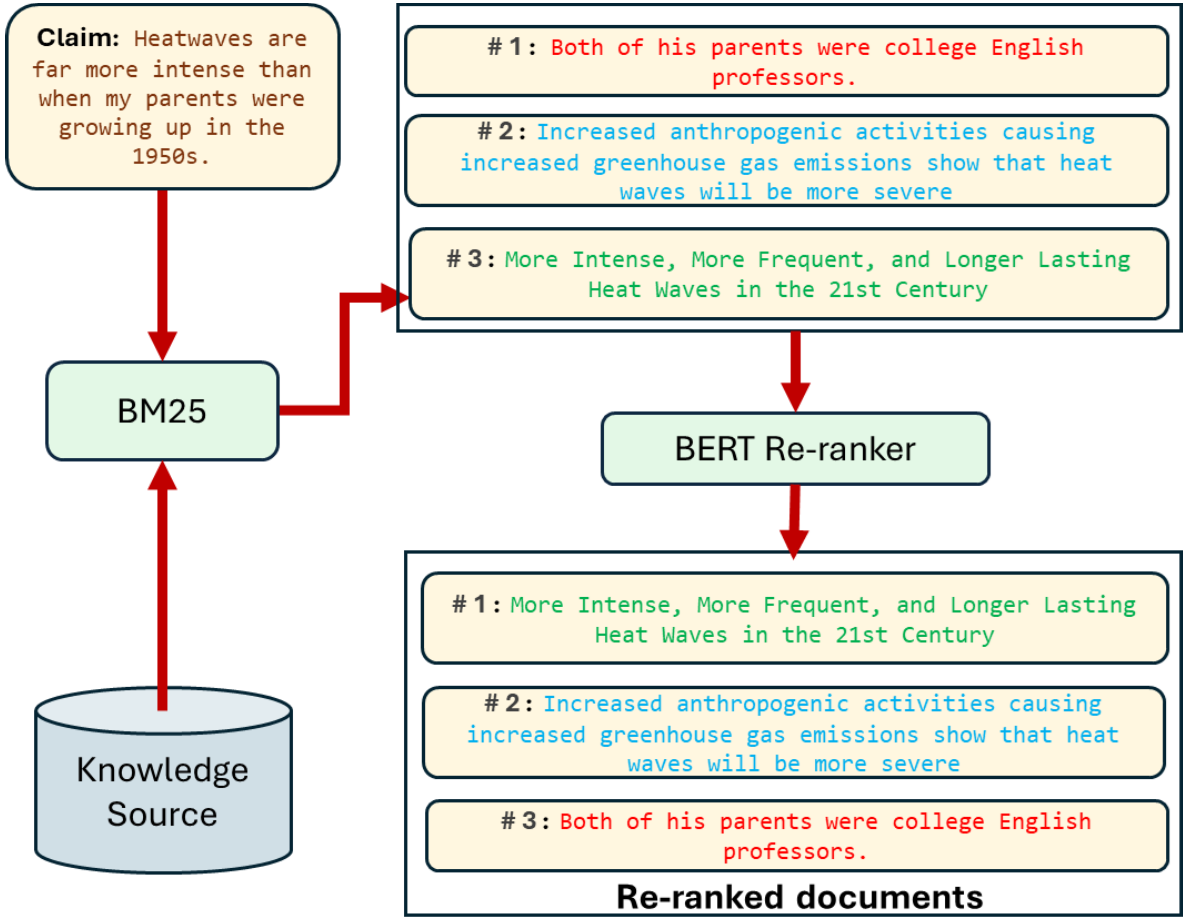
Retriever: BM25 with BERT Re-ranking

- Can use a **pre-trained** BERT model to **re-rank** the documents retrieved by BM25
Nogueira and Cho (2019).

- Can significantly improve both **recall** and **precision.**

- Jointly encode claim and document, then perform binary classification

[[CLS], **CLAIM**, [SEP], **DOC**, [SEP]]



Retriever: BM25 with BERT Re-ranking

Re-ranking really **works best** when we set a large k (e.g. $k = 1000$) and retrieve the top-k documents using BM25, then re-rank and keep the top-k' with $k' << k$ (e.g. $k' = 10$).

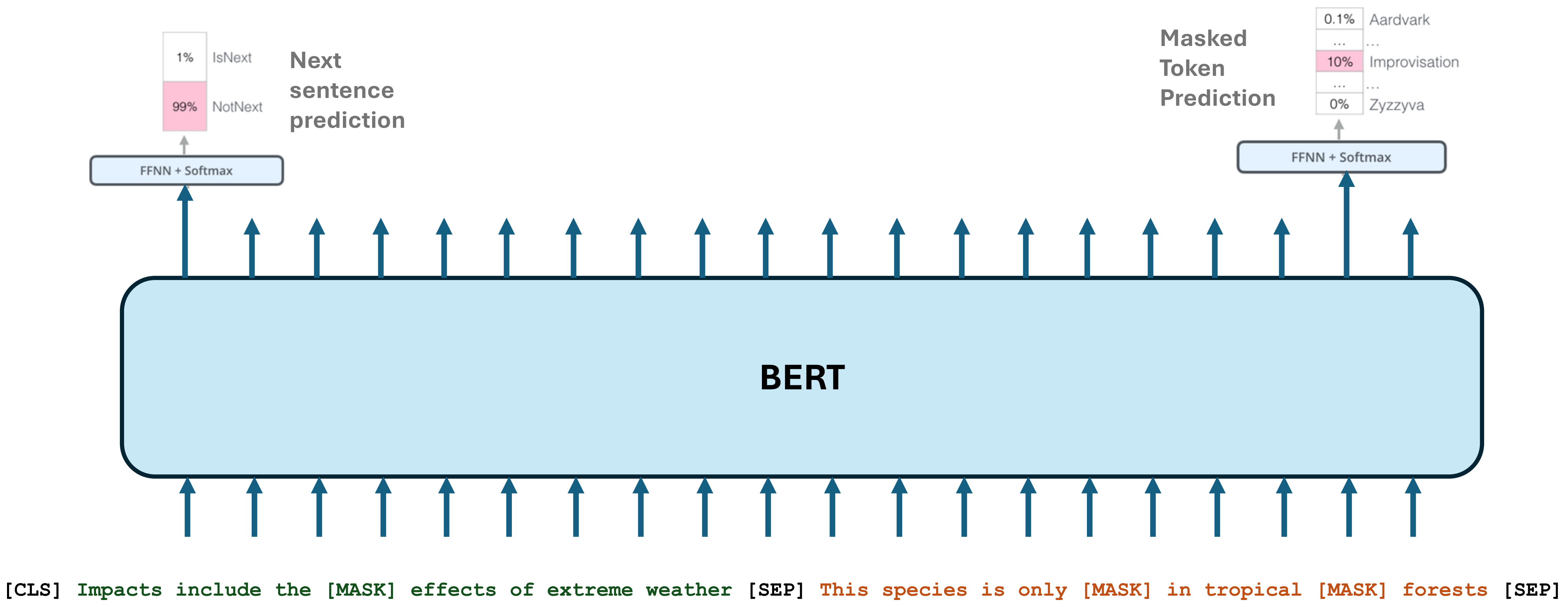
Demo:

```
claim-2692: Volcanoes, solar variations, clouds, methane, aerosols - these all change the way energy
enters and/or leaves our climate.

Gold evidence passages: ['evidence-139375', 'evidence-927438', 'evidence-58290’]

evidence-139375 --> BM25 Rank: 422, After Re-ranking: 32
evidence-58290  --> BM25 Rank: 86,  After Re-ranking: 42
evidence-927438 --> BM25 Rank: 350, After Re-ranking: 5

Precision: 0.003, Recall: 1.0, F1: 0.005982053838484547
```

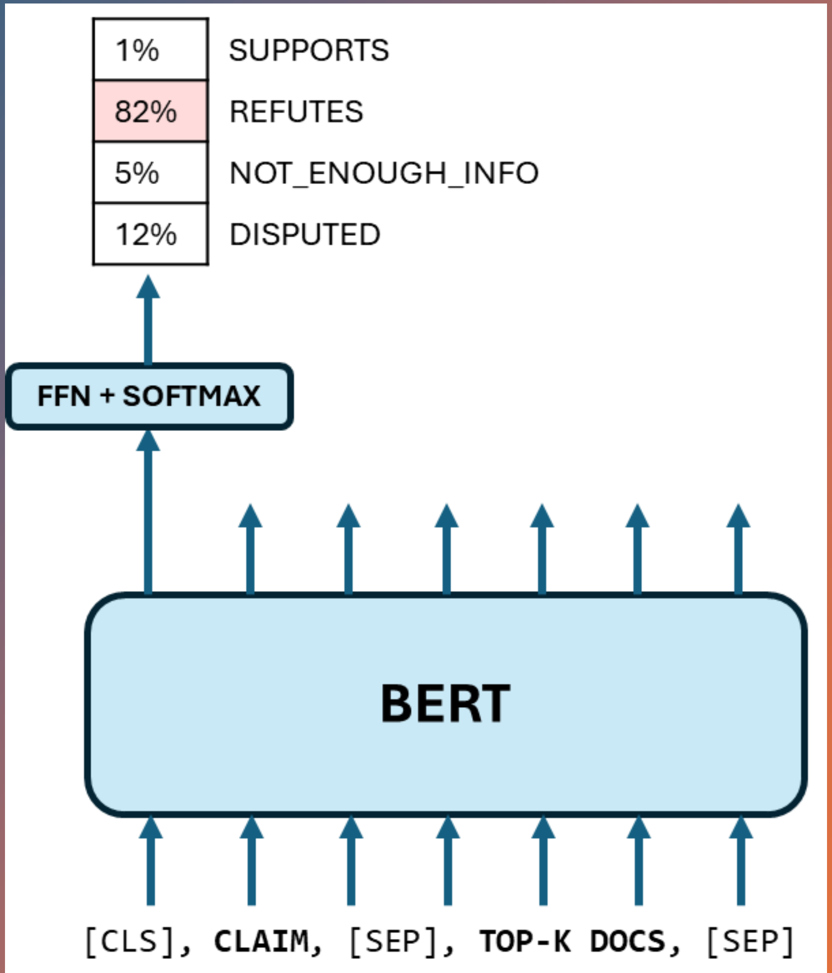
Transfer Learning and custom BERT

- **Finetuning** a pretrained language model on downstream tasks with small datasets proven to work extremely well.
- Implement and **pre-train** our own **custom BERT** and **WordPiece** tokenizer!
- Pre-training dataset contains **sentence pairs**: (claim, evidence) and (evidence, evidence). Next sentence prediction labels are assigned according to gold evidence list.
- Jointly trained on Masked Language Model and Next Sentence prediction tasks.
- Identical model architecture and similar hyperparameters as original BERT, except we use 512 embedding dims and 8 encoder layers instead of 12.

Transfer Learning and custom BERT



About 15% of tokens in the input sequence are randomly masked.



Classifier

- Finetune custom BERT for claim classification.
- Input is a single sequence containing claim and top-k concatenated re-ranked documents.
- Attach an extra “classifier head” which is just a linear layer that map [CLS] embedding into output logits.
- Currently exploring different values of k and thresholding schemes.

Recap: Summary of Our Approach

Training:

1) Pre-train custom BERT on Masked Language Model and Next Sentence Prediction Tasks Jointly

2) Train a BM25 model

3) Finetune custom BERT on re-ranking task

4) Finetune custom BERT classifier on claim classification

Inference:

1) Given a claim, use BM25 to retrieve top-k documents

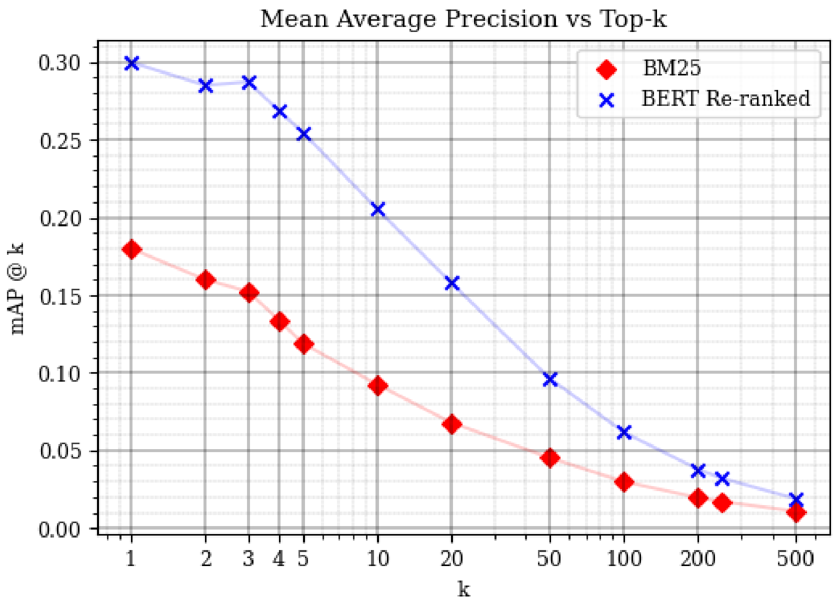
2) Apply BERT re-ranking and filter top-k’ documents (k’ << k)

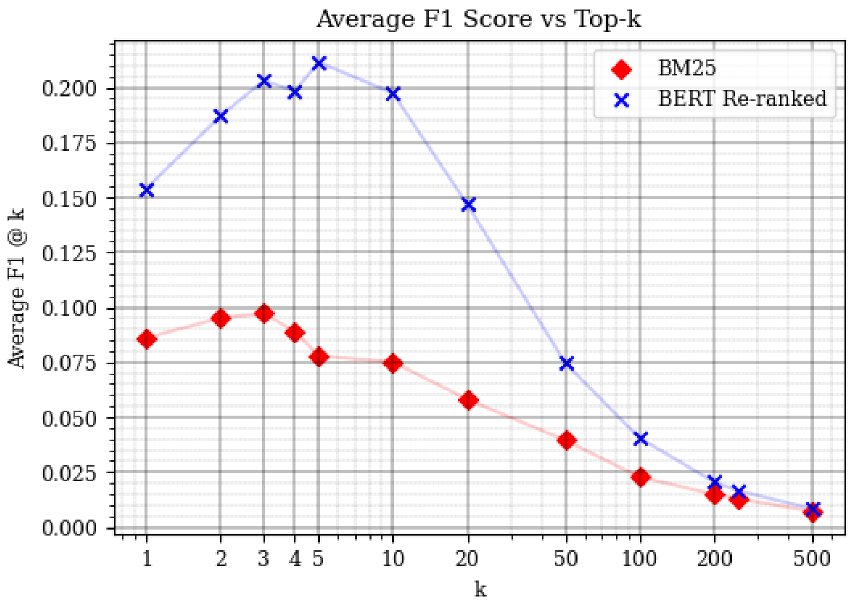
3) Classify with input containing claim and concatenated top-k’ documents

Preliminary Results

Retriever Performance

After finetuning the pre-trained custom BERT model for re-ranking, we observe substantial improvement in retrieval performance on the dev set.





	BM25	BM25 + BERT Re-ranking
Average Recall@5	10.7%	31.2%
Average F1@5	7.8%	21.2%
mAP@5	11.9%	25.5%

Conclusions and Future Work

Main Limitations of Our Approach

Pretraining on our small dataset and knowledge source => risk of overfitting

Did not have access to open-source pre-trained transformer language model

Did not have access to publicly available fact-verification datasets for training our system

Computational resource constraints => had to use smaller model, not able to perform extensive hyperparameter tuning on transformer model or train for long periods of time

Did not explore data augmentation techniques, such as paraphrase generation via backtranslation, due to computational resource constraints

Models don't incorporate "hand engineered" features, external domain knowledge or any rule-based techniques, purely machine learning driven

Conclusion and Future Work

- Automated Fact-Verification is a challenging task and requires Natural Language Understanding (NLU)!
- Transformer based language models are highly effective, but **small models pre-trained on small datasets not enough to attain NLU capability**

- Need to investigate the effects of pre-training larger model on much larger open corpus.
- Need to investigate other transformer architectures, such as transformer decoder and seq2seq.
- Need to investigate effects of jointly training on multiple tasks related to claim-verification and see if possible to enhance performance.