

Multilingual and Cross-lingual Fact-Checked Claim Retrieval

<https://disai.eu/semeval-2025/>

Multilingual and Cross-lingual Fact-Checked Claim Retrieval

Overview

The Task is to efficiently identifying previously fact-checked claims (Annotated by multiple Experts) across multiple languages . We will **develop systems to retrieve relevant fact-checked claims** for given social media posts, using the MultiClaim dataset of over 200,000 fact-checked claims and 28,000 social media posts in 27 languages . Evaluating the System in terms of two metrics, mean reciprocal rank and success-at-K in monolingual and cross lingual separately .

The primary goal of the project would be to implement a good model for English monolingual setup and extending it to multilingual setup if time and resources permit .

The project has significant potential for real-world impact in the fight against misinformation .

Dataset

The dataset used will be MultiClaim dataset , which have over 200,000 fact-checked claims and 28,000 social media posts in 27 languages , also the the translations of different languages into English is present through Google translate API which can be used for English monolingual setup.

The sample data consists of three different parts:

1. **fact_checks** -

This file contains all fact-checks. It has four fields:

- fact_check_id
- claim – This is the translated text (see below) of the fact-check claim, original text is also contained.
- instances – Instances of the fact-check – a list of timestamps and URLs.
- title – This is the translated text (see below) of the fact-check title

2. **posts-**

This file contains all social media posts. It has five fields:

- post_id
- instances – Instances of the fact-check – a list of timestamps and what were the social media platforms.
- ocr – This is a list of texts and their translated versions (see below) of the OCR transcripts based on the images attached to the post.
- verdicts – This is a list of verdicts attached by Meta (e.g., False information)
- text – This is the text and translated text (see below) of the text written by the user.

3. **fact_check_post_mapping**

This file contains the mapping between fact checks and social media posts. It has three fields:


- fact_check_id: the id of the fact check in the fact_checks
- post_id: the id of the post in the posts
- pair_lang: the language info about this mapped pair. For example, spa-eng stands for SMP in Spanish and FC in English.

Literature Survey and related works

1. Exploring Dual Encoder Architectures for Question Answering

Exploring Dual Encoder Architectures for Question Answering

Dual encoders have been used for question-answering (QA) and information retrieval (IR) tasks with good results. Previous research focuses on two major types of dual encoders, Siamese Dual

 <https://arxiv.org/abs/2204.07120>



This paper explores dual encoders models ability to retrieve correct answers . the dataset is used is MS MARCO, open domain NQ and the MultiReQA benchmarks

2. Retrieval-Enhanced Dual Encoder Training for Product Matching

aclanthology.org

<https://aclanthology.org/2023.emnlp-industry.22.pdf>

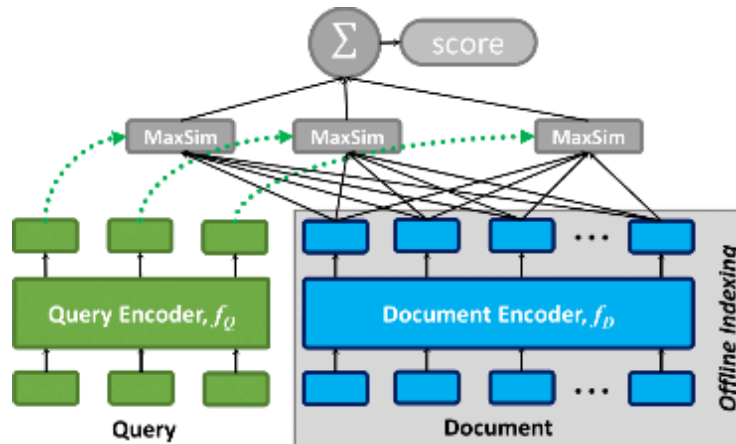
The Product matching is the task of finding the same product in a catalog for a specific query product. Dual encoders had been proposed as a state-of-the-art solution

3. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT

<https://arxiv.org/pdf/2004.12832>

we present ColBERT, a novel ranking model that adapts deep LMs (in particular, BERT) for efficient retrieval. ColBERT introduces a late interaction architecture that independently encodes the query and the document using BERT and then employs a cheap yet powerful interaction step that models their fine-grained similarity

https://huggingface.co/sebastian-hofstaetter/colbert-distilbert-margin_mse-T2-msmarco



Structure

Experimental Details

Data Splits: We Divide the data into 80:20 split for train and validation set .we shuffle the data to avoid any biases .

Hardware Setup - We had access to ADA with 4 GPU'S

Pre-Processing

1. English Translations were used for different language data .
2. The pre-processing steps included multiple regex functions to replace twitter_links, elongations for better encodings.
3. Lemmatisation was done using porter stemmer for Word matching Based Algorithm like BM25

Approaches

1. **Base lining** -

BM25

- We used BM25 API as the data was very large for to compute and python data structures were failing to calculate .
- Results BM25 -

success@10	0.573456674
success@50	0.6898923
success@100	0.7912348

This Result Shows that word matching is not a very good method because the dataset contains many claims which are very similar to each other in terms of vocabulary .

Hyper-parameters: K-1.25,b-0.75

2. Embedding Models

- a. Sentence encoders like 'all-MiniLM-L6-v2' and 'all-mpnet-base-v2' were used to encode all claims and were stores in a FAISS vector database .
 - **all-MiniLM-L6-v2:** The output dimension is 384 ,It has approximately 44 million parameters.
 - **all-mpnet-base-v2:** The output dimension is 768,It has approximately 110 million parameters.
- b. Queries were encoded with the same sentence encoders and similarity score was generated using cosine similarity.
- c. top k claims were retrieved based on similarity score
- d. Results:

Model	success@10	success@50	success@100
all-MiniLM-L6-v2	0.6323352	0.7223489	0.7312330
all-mpnet-base-v2	0.653462345	0.743210	0.7878431

3. Colbert Pre-trained

- a. We used pr-ettrained colbert model from hugging face "sebastian-hofstaetter/colbert-distilbert-margin_mse-T2-msmarco" which was previously trained for similar task of ranking documnets based on queries .
- b. Results observed were marginally better than all-mpnet-base-v2
- c. Hyperparameters -

Model	success@10	success@50	success@100
Colbert pre-trained	0.67	0.72	0.81

4. Dual-Encoder Architecture

Hyperparameters

Ensemble Models

We tried different combinations of the above models to reduce the number of possible samples from 200,000 to a lower number .

BM25 + all-MiniLM-L6-v2:

- we sampled top 10000 claims using BM25 which significantly reduced the search space for all-MiniLM-L6-v2 embeddings model
- We did not observe any significant improvement in the model performance compared to direct all-MiniLM-L6-v2 usage .
- We concluded that all-MiniLM-L6-v2 is already a good model and BM25 is not adding much .

all-mpnet-base-v2 + Dual-Encoder Architecture

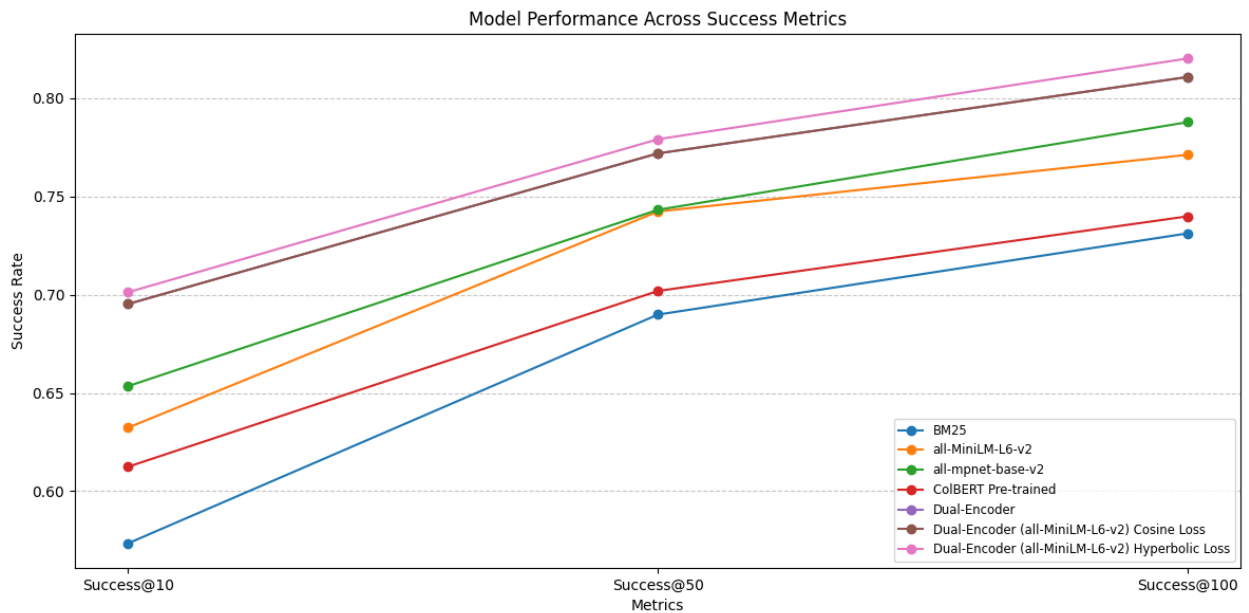
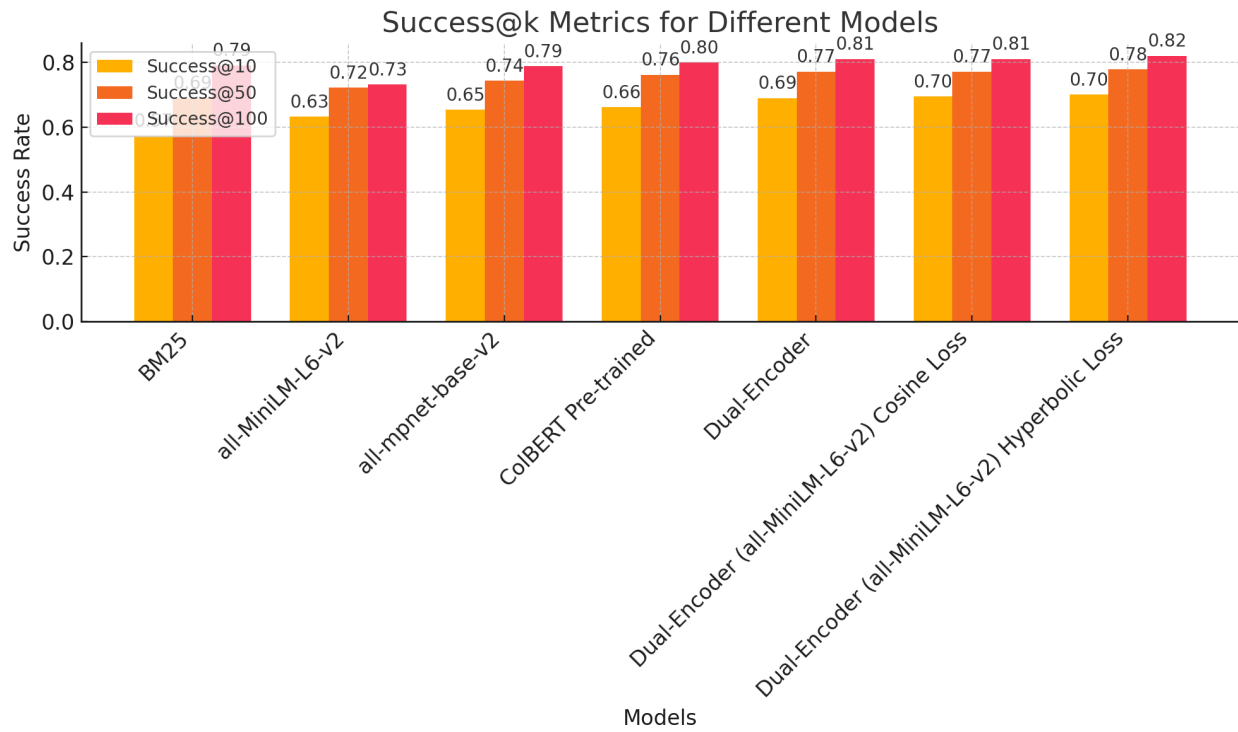
- top 1000 claims were retrieved using all-mpnet-base-v2 model and then dual-encoder was used to retrieve top k (10, 50 ,100).

Ranking the approaches

The approaches tried would be tested on Common metrics including success@k (which is the accuracy of model of retrieving the correct fact in the top k searches) .

Models	Success@10	Sucess@50	Sucess@100
BM25	0.573456674	0.6898923	0.7312348
all-MiniLM-L6-v2	0.6323352	0.7423489	0.7712330
all-mpnet-base-v2	0.653462345	0.743210	0.7878431

Colbert Pre-trained	0.6124384	0.7019234	0.73991273
Dual -Encoder(all-MiniLM-L6-v2) cosine loss	0.69526346	0.7719283	0.81081723
Dual -Encoder(all-MiniLM-L6-v2) hyperbolic loss	0.70124564	0.779134	0.820245



Error Analysis

- The dataset contains claims which are very lexically and semantically very similar to each other . these examples create ambiguity and some claims have 100's of similar claims .
- eg . images of red sky in china are real.',
'the sky of china is red in a strange phenomenon',
'picture of the sky in china turning red.',
'the sky turned blood red in china',

Scalability

- We have saved the models on one-drive and built a gradio-app which can be used to retrieve the top 10 documents given a query .

Ablation Study

- We tried different pre-processing steps to see the results including lemmatisation , stemming . but the results were quite low for embeddings based models .
- We tried many different sentence encoders like bert-base-uncased , roberta .
- We tried ensemble models to get to get better latency . but since the models were saved in vector database ensemble models reduced success@k scores and did not have large impact on latency .

Future Work

- Extension of this project to multilingual and cross lingual setup .