PRODUCT RETRIEVAL ON E-COMMERCE - GROUP 18

Dao Trong Viet
Dao Van Tung
Bui Thanh Tung
Nguyen Anh Tuan
Nguyen Phuong Uyen

ABSTRACT

This project presents a novel dataset and a hybrid framework for product retrieval on e-commerce platforms. The dataset consists of products and queries in Vietnamese, labeled by combining automated and manual methods. The hybrid framework combines bi-encoder, cross-encoder, and BM25 models to enhance the retrieval accuracy and relevance. The experimental results show that the proposed approach outperforms individual models and achieves stable performance across various metrics. However, compared with other combining methods, there is no best model, suggesting the unstable of retrieval systems. The paper also discusses the challenges and limitations of the current methods and suggests directions for future research. The code is available at GitHub.

1 INTRODUCTION

In the dynamic realm of e-commerce, where many products are available across online platforms, the efficient retrieval of desired items has become a central focus. The "Product Retrieval on E-Commerce" project centers on the development of innovative methodologies and systems tailored to enhance search functionalities within e-commerce platforms.

Our dataset comprises product type information, spanning multiple types, and detailed product descriptions. Unlike conventional approaches, we do not incorporate user reviews or shopping trends in our analysis. The challenge lies in extracting meaningful insights and improving retrieval accuracy solely based on the product type and description, showcasing a unique and focused approach.

To address this specific scenario, our research explores advanced techniques such as utilizing probabilistic information retrieval (BM25) and a classic term-weighting scheme (TF-IDF). By leveraging the inherent characteristics of product types and descriptions, we aim to design a system that excels in delivering precise search results, meeting user expectations without relying on additional user-generated content.

The significance of this project extends beyond simplifying the consumer's search experience; it also contributes to the optimization of e-commerce operations without recourse to external factors like reviews or trends. Through the streamlined retrieval process, businesses can anticipate improved efficiency and user satisfaction in navigating the diverse products offered on e-commerce platforms.

Our contributions can be summarized as follows:

- We curated and labeled a comprehensive training and test dataset by sourcing data from the internet, employing both automated and manual annotation methods. This method is designed to minimize human intervention and maximize the usage of AI tools like ChatGPT.
- We thoroughly evaluated various methods in the product retrieval problem, including TF-IDF, BM25, and Hybrid search.
- We fine-tuned a Vietnamese XLM-Roberta model, serving as the initial checkpoint for the cross-encoder model.
- We propose an effective training procedure for both bi-encoder and cross-encoder models, addressing challenges associated with limited data and overconfidence.
- We translated a question-answering dataset that is included in training semantic models to help them understand natural language queries.

2 Related Works

With the proliferation of online shopping, the challenge of effectively searching for products on e-commerce platforms has garnered significant attention from researchers. Despite the growing prevalence of natural language usage on the internet, most search engines still rely on key-based systems for matching. In recent years, the application of deep learning to learn semantic representations has become a prominent area of investigation.

Several widely adopted methods include Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, Bidirectional Long Short-Term Memory (Bi-LSTM) networks, and Transformers Vaswani et al. (2023). Reimers et al. Reimers & Gurevych (2019a) proposed two distinct models—one for extracting sentence features and another for estimating the similarity between two sentences. Their work demonstrated that Transformers are state-of-the-art (SOTA) methods capable of learning contextual information and providing superior representations compared to other methods such as GloVe Pennington et al. (2014).

In addition, Li et al. Li et al. (2021) introduced frameworks that combine multiple models and user history to enhance overall user experiences. However, it is noteworthy that their focus encompasses not only the content of the user's query but also incorporates user preferences. In contrast, our work concentrates solely on extracting relevant information from user queries without explicitly addressing user preferences.

3 Dataset

3.1 Data Collection

In the data collection phase of this project, we employed a Python script leveraging the requests library to interact with the Tiki.vn e-commerce platform's API. This process involved systematically retrieving product IDs by navigating through multiple pages of the Tiki.vn API, with each page containing a predefined set of product listings. The collected product IDs were then organized into a pandas DataFrame and saved to a CSV file. Subsequently, we executed a comprehensive data retrieval process, obtaining detailed product information for each identified product using its corresponding ID. The script utilized cookies to simulate an authenticated user session, ensuring access to the API. To prevent potential server overload, random delays of 3 to 10 seconds were introduced between consecutive requests. The parsed information, encompassing product names, short descriptions, and other pertinent details, was meticulously stored in a new pandas DataFrame. The final dataset, encapsulating both product IDs and detailed information, was saved in a CSV file. This meticulously curated dataset forms the foundation for subsequent analysis and model development in our pursuit of optimizing product retrieval on e-commerce platforms.

3.2 Product retrieval dataset

To the best of our knowledge, as of the present moment, there exists no Vietnamese dataset specifically tailored for the product search domain. While numerous datasets pertinent to this domain exist in various languages, a majority of them remain privately held and pose challenges in terms of accessibility. In the present study, we advocate for the creation of an entirely novel dataset designed to assess retrieval systems within the e-commerce domain. Given constraints in terms of time and human resources, we employed a two-fold methodology involving initial automated procedures followed by subsequent meticulous manual re-labeling. The comprehensive procedural framework is visually elucidated in Figure 1.

In the beginning, we created a set of 100 seed questions and selected 1000 products from the gathered dataset. We apply methods such as hybrid search, BM25, TF-IDF to retrieve candidate products for each question. Particularly, each method will recommend 3 products for each question. After that, we use ChatGPT to generate for each question in the seed set 10 similar questions. At the end of the process, we have a dataset that contains 1000 queries and their related products. This dataset will be used to evaluate our proposed methods in the next section.

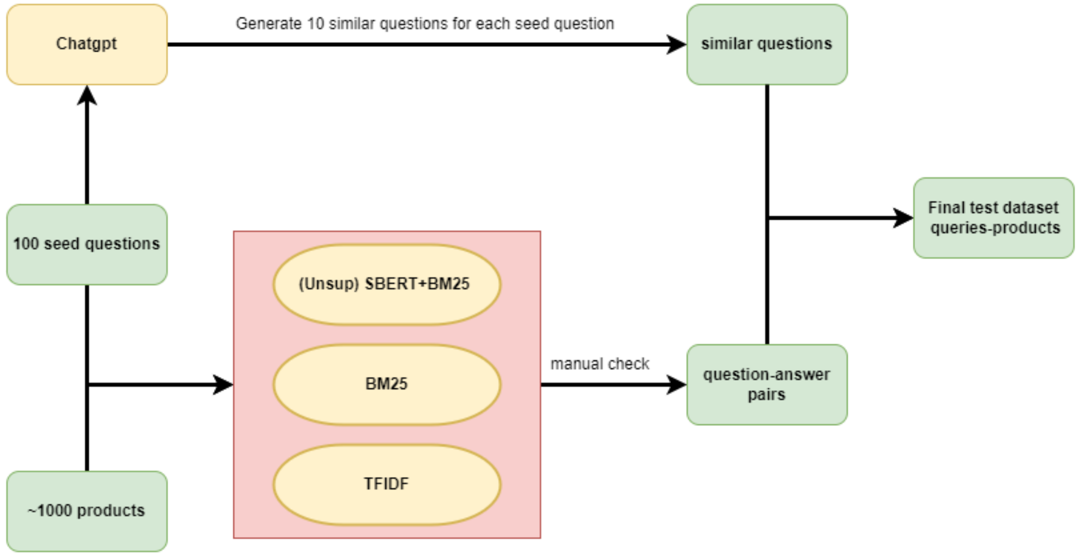

Figure 1: Data Labeling Process

4 Approach

4.1 Problem formulation

Given a query s, a set of n products Q, and a model θ are used to estimate the similarity between the query and product descriptions. The objective of product search system is to search correctly product p for query s:

$$p = \arg\max(\theta(s, p_i)), i \in [1; n]$$

4.2 TF-IDF

Inverse document frequency (IDF) is a fundamental and widely employed concept in the realm of information retrieval, often paired with term frequency (TF) to create an effective term weighting scheme for information retrieval systems. The essence of TF-IDF lies in gauging the relative frequency of words in a specific document compared to their inverse proportion across the entire document corpus. This calculation intuitively reveals the relevance of a given word within a particular document, assigning higher TF-IDF values to words that are less common across the entire corpus.

The formal implementation of TF-IDF involves a consistent approach: given a document collection D, a word w, and an individual document $d \in D$, the calculation is performed as follows:

$$w_d = f_{w,d} \cdot \log\left(\frac{|D|}{f_{w,D}}\right)$$

Here, $f_{w,d}$ represents the frequency of word w in document d, $|D|$ is the size of the corpus, and $f_{w,D}$ is the number of documents in which w appears in D.

The resulting wd reflects the importance of the word w in the context of the specific document d. For instance, common words across a small set of documents receive higher TF-IDF scores, emphasizing their significance within those specific documents. On the other hand, extremely common words, like articles and prepositions, are assigned lower TF-IDF scores, making them negligible in searches.

The code for implementing TF-IDF is elegantly simple. For a given query q comprising words w_i, wi, d is calculated for each w_i in every document $d \in D$. The calculation involves iterating through the document collection, maintaining running sums of $f_{w,d}$ and $f_{w,D}$. Once these sums are obtained, wi, d is calculated based on the established mathematical framework. The final step involves returning a set D^* containing documents d to maximize the following equation:

$$\sum_i w_{i,d}$$

The size of D^* can be determined either by the user or the system, and the documents are returned in descending order according to the calculated values. This adheres to the traditional method of implementing TF-IDF.

4.3 BM25

BM25 functions as a ranking function in information retrieval, seeking to identify the most pertinent document in response to a given query. Its methodology involves computing a ranking score using term frequency, inverse document frequency, and document length:

- **Term frequency (TF)**: This represents how frequent a term appeared in the document.
- **Document length (DL(D))**: The length of a document D
- **Inverse document frequency (IDF)**: $IDF(t) = log(\frac{N}{count(t)})$ where N is total number of document while $count(t)$ is number of document containing the term t. This measure how distinctive and rare a term is across every document; thus, assigning more weight toward infrequent term.
- **Hyperparameter k and b**: While k influences how responsive the overall score to TF, b controls the impact of DL

The formulation of the BM25 score can be written as follows:

$$BM25(D, Q) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{TF(q_i, D) \cdot (k+1)}{TF(q_i, D) + k \cdot (1 - b + b \cdot \frac{DL(D)}{Avg\_DL})}$$

- D: The document we are considering
- Q: The query we are trying to score.
- n: The number of terms in Q
- q_i: The $i - th$ query term

Following the computation of scores for each combination of Q (query) and D (document) using the BM25 algorithm, our next step involves selecting the top three documents that exhibit the highest scores about the given query Q. In other words, we aim to identify and retrieve the three documents that are most relevant to the query based on their BM25 scores.

4.4 Hybrid Search

4.4.1 Transformer

Transformer architecture was first introduced in Vaswani et al. (2023) and now become the dominant model in both natural language processing areas. Transformer is a language model that formulates NLP problems in a sequence-to-sequence fashion or conditional probability as the following:

$$P(Y|X, \theta) = \prod_{t=1}^{T} p(y_t|X, \theta)$$

where $Y = (y_1, y_2, ..., y_t)$ is the target sentence, X is the source sentence, θ is the language model.

The detailed architecture is presented in figure 2. The model contains two main blocks: encoder and decoder. Both of them have quite similar components. The encoder model is a sequence of attention and feed-forward layers. The input text is first tokenized to tokens by using algorithms such as BPE Sennrich et al. (2016). The representation of tokens is looked up from the embeddings

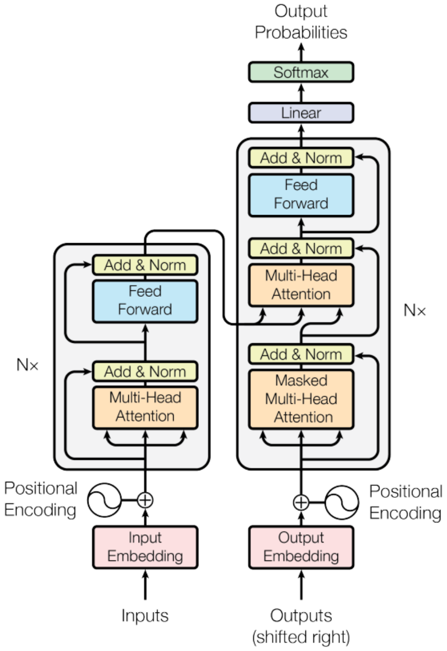

Figure 2: Transformer Vaswani et al. (2023)

layer and then summed up with position embeddings (that encode position information). In the next steps, representations are fed into the attention layer. These features will play roles of query, key, and value vectors. A token will query other tokens to get contextual information:

$$Q = W_q \times x$$
$$V = W_v \times x$$
$$K = W_k \times x$$
$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

where d_k is the dimension of queries and keys.

The resulting features will be fed into the feed-forward layer to learn more complex features. This process will be repeated many times, making it contain high semantic and contextual information. To avoid gradient vanishing, as well as, stabilize the training process, residual connection and normalization layer are also applied to the model.

The Decoder block is like the encoder but has attention layers to query information from the encoder. One more key difference is that the decoder is designed for causal generation, which means that a token can only attention to its previous tokens.

Encoder-only models such as BERT Devlin et al. (2018) or RobertaLiu et al. (2019) are pre-trained on predicting masked words task, and later can be used for classification task or extract sentence representation. Particularly, for each sentence, a rate of words will be masked, the model needs to read the context and then predict the missed words. Previous show that this objective helps the model to learn to extract meaningful contextualized representations.

4.4.2 Sentence Transformer

Reimers & Gurevych (2019b) showed that a Siamese model based on the encoder of the transformer can be used to extract sentence representation. Figure 3 shows the scheme of the bi-encoder model. We use an encoder to encode consecutively two sentences A and B. The output features will be fed

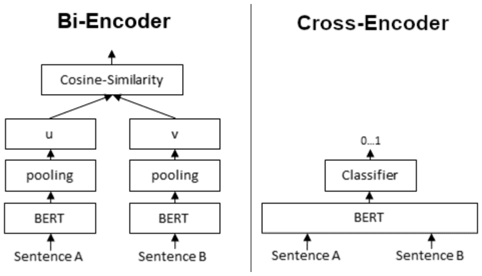

Figure 3: Sentence Transformer Reimers & Gurevych (2019b)

into a mean pooling to get features u and v, respectively. The cosine distance is used to measure the similarity between A and B. In many real scenarios, the similarity scores are hard to gather. In this work, we have only positive pairs (product name and product description), thus the contrastive strategy is applied. The positive pairs are created from a pair of (product name and product description) or (two identical sentences). In each mini-batch, we have set of pairs $[(x_1, y_1), (x_2, y_2), ..., (x_k, y_k)]$.

$$loss = -\frac{1}{K}\sum_{i=1}^{K}\left[-\log\frac{e^{\theta(x_i,y_i)}}{\sum\limits_{j=1}^{K} e^{\theta(x_i,y_j}}\right]$$

where θ is the bi-encoder model.

An alternative method for estimating similarity scores involves the utilization of a cross-encoder model. In this approach, the sentences A and B are concatenated and then fed into the encoder to predict the score. The advantage lies in the model's ability to simultaneously consider both sentences, leading to more accurate score generation. However, this method necessitates merging the sentences, resulting in increased search times when dealing with large databases. In practical applications, this model is often employed to reevaluate results obtained from bi-encoder models.

During the training process, akin to the bi-encoder, only positive pairs are considered, with negative pairs randomly sampled from the corpus. A notable challenge with the cross-encoder model is the issue of overconfidence. To mitigate this, the scores for positive pairs are randomly selected from the range $[0.7, 1]$, while the scores for negative pairs range from 0 to 0.3.

Nevertheless, utilizing only titles and descriptions as positive pairs may not accurately reflect real-world scenarios, where users typically pose questions in natural language rather than using keywords. To address this discrepancy, during the training phase of both bi-encoder and cross-encoder models, we augment the dataset with questions and their corresponding answers from the Quora dataset, translated into Vietnamese using the Google API. This dataset comprises question-answer pairs from the Quora website, offering a suitable foundation for constructing a natural query-based retrieval system.

4.4.3 Hybrid Framework

In common pipelines, a query will go through a bi-encoder to search a small set of related products, and then rerank by the cross-encoder. The final results will be the top k products. However, semantic methods usually are hard to interpret and may return unexpected results. To alleviate this issue, in the proposed phase (before the reranking step), we add results from BM25 algorithm that are based on related keywords. As shown in qdr BM25 scores and bi-encoder scores are not correlated. Hence, we merge both results of these models and use the cross-encoder to rerank.

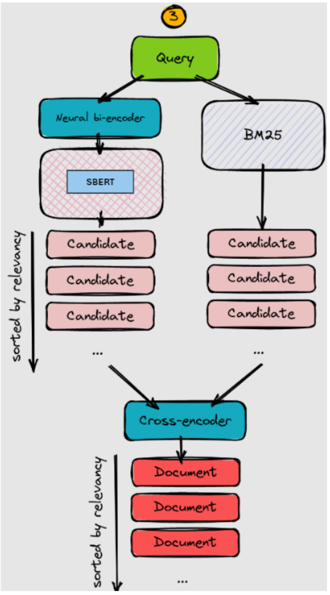

Figure 4: Hybrid Search Framework qdr

5 Experiment Results

5.1 Experiment Settings

5.1.1 Dataset

To evaluate our approaches, we performed experiments on the dataset we assembled, labeling the ground truth manually. For every test query, the ground truth consists of a set of product IDs that are most pertinent to that particular inquiry. These queries are matched against a table containing product information, including ID, name, and description.

There are a total of 360 queries with lengths ranging from 23 to 98 terms. The database we aim to query contains almost 1000 unique products. The product descriptions vary in length from 5 to 200 terms, while the product names range from 51 to near 250 characters.

5.1.2 Metrics

- **Precision@k**: Precision@K is a metric used to evaluate the precision of a retrieval system by considering the relevance of the top K items returned. The formula for Precision@K is given by:

$$\text{Precision@K} = \frac{\text{Number of relevant items in top } K}{K}$$

  In the context of our work on product retrieval in e-commerce, this metric quantifies the accuracy of the system in presenting relevant products within the top K positions of the ranked list. For example, if Precision@5 is 0.8, it indicates that 80% of the top 5 products retrieved by the system are relevant to the user's search query.

- **Average Precision**: Average Precision provides a more nuanced evaluation by considering precision at each relevant item position in the ranked list and then averaging these values. The formula for Average Precision is as follows:

$$\text{Average Precision} = \frac{\sum_{k=1}^{n} (\text{Precision at position } k \times \text{Relevance at position } k)}{\text{Total number of relevant items}}$$

This metric assesses how well the system maintains precision across all relevant products in the retrieval process. It captures the overall quality of the ranking order by giving higher weight to relevant items appearing at higher positions in the list.

- **Mean Average Precision (MAP)**: Mean Average Precision builds upon Average Precision by taking the average of Average Precision values across multiple queries. The formula for Mean Average Precision is given by:

$$\text{MAP} = \frac{\sum_{q=1}^{Q} \text{Average Precision for query } q}{Q}$$

MAP provides a comprehensive measure of the system's performance across different search queries. A higher MAP value indicates that the system consistently provides accurate and relevant results across a diverse set of user queries.

5.1.3 Baseline Models

- **TF-IDF**

  We start by processing our dataset, which comprises product name and detailed product descriptions. First, we tokenize the product descriptions to break them into individual words or tokens. After that, commonly occurring words that carry little semantic meaning, known as stop words, are removed from the tokenized product descriptions.

  Next, we use the sklearn library for the implementation of TF-IDF. The TF-IDF vectorizer is applied to the preprocessed product descriptions to convert them into numerical vectors. This transformation considers the importance of each term in the entire corpus, providing a numerical representation that emphasizes the unique characteristics of each product description.

  To further enhance the efficiency of our system and reduce the dimensionality of the TF-IDF matrix, we apply Singular Value Decomposition. The dimensionality reduction is achieved by retaining only the top-k singular values and their corresponding vectors.

  The TF-IDF matrix, now reduced in dimensionality through SVD, will serve as the input to our retrieval model.

- **BM25**

  Initially, we eliminate punctuations from each document and break it down into individual words or tokens. Employing the processed dataset as our corpus, we precompute the inverse document frequency, term frequency, and document length, storing these values in a Python dictionary. During the inference phase, we utilize this dictionary to calculate the BM25 score.

- **Bi-encoder only** In this setting, we use only bi-encoder to get top k products without reranking again.
- **Cross-encoder only** The query is concatenated to the description and get top k highest score.
- **Bi-cross only** Bi-encoder proposes a list of products and the cross-encoder reranks results. There is not proposal from the BM25 algorithm.
- **Bi-BM25** Bi-encoder proposes a list of products and the BM25 reranks results. There is not proposal from the BM25 algorithm.
- **Hybrid Search** the bi-encoder model is fine-tuned from checkpoint of BKAI lab [1] and the cross-encoder is fine-tuned from XLM-Roberta [2]. The optimizer is AdamW with a learning rate from $1e-5$ to $2e-5$. Models are trained on Nvidia P100 and implemented in the PyTorch framework.

5.2 Results

The experiment results are presented in Table 1. It is evident that no single model excels across all metrics. Notably, the Bi-encoder only model achieves the highest precision@1 at 33.89%. How-

[1] https://huggingface.co/bkai-foundation-models/vietnamese-bi-encoder

[2] https://huggingface.co/xlm-roberta-base

	$P@1$	$P@5$	$P@10$	MAP
TF-IDF	30.28	20.83	**16.25**	25.66
BM25	25.56	21.39	15.83	21.50
Hybrid	32.50	**22.38**	15.69	38.41
Bi-encoder only	**33.89**	21.22	14.94	39.32
Cross-encoder only	28.61	19.61	13.03	35.08
Bi-cross only	32.27	21.89	14.94	40.34
Bi-BM25	30.55	21.11	14.94	**40.44**

Table 1: Results on the test dataset

ever, its precision at other top-k and MAP values are lower compared to other methods, suggesting potential ranking inconsistencies. The proposed Hybrid model attains the highest precision@5 with 22.38%. While not state-of-the-art in all metrics, its performance closely rivals the best models, with gaps of 1.39%, 1.16%, 0.56%, and 2.03% at precision@1, precision@5, precision@10, and MAP, respectively. Compared to other methods, the Hybrid model demonstrates stable and reliable performance across various metrics.

The Hybrid model combines the Bi-encoder, Cross-encoder, and BM25 models. The ablation study in Table 1 demonstrates that the proposed model improves overall results compared to individual models. The MAP of the Hybrid model surpasses that of BM25 and Cross-encoder only, but falls short of Bi-encoder only, Bi-cross only, and Bi-BM25. In terms of precision@5 and precision@10, the Hybrid model outperforms its sub-models. Notably, it outperforms Bi-cross only in precision@1, suggesting that BM25 contributes relevant products to the proposed list of Cross-encoder models. Furthermore, the Bi-cross only model outperforms Bi-encoder only and Cross-encoder only, indicating the effectiveness of combining these models to enhance the precision of the retrieval system.

6 Conclusion

In this study, we introduce a novel test retrieval dataset tailored for the e-commerce domain and put forth a sophisticated hybrid framework that amalgamates multiple models, including cross-encoder, bi-encoder, and BM25. Our experimental findings substantiate the robustness of our proposed methodology, showcasing its reliability in comparison to standalone methods such as TF-IDF, BM25, cross-encoder only, and bi-encoder only. Notably, our proposed approach demonstrates scalability for real-world applications, with the potential to significantly enhance user experience through natural language queries. Furthermore, our introduced dataset emerges as a potent benchmark, offering a robust foundation for evaluating the efficacy of retrieval systems. By proposing in retrieval system performance evaluation, our work contributes to the broader landscape of information retrieval research, making it a valuable resource for researchers and practitioners alike.

7 Limitations

As evident from the experimental results, the cross-encoder model exhibits a degree of unreliability. While contributing to improvements in certain metrics, it concurrently leads to declines in others. This observation underscores the need for further refinement in the training process of the cross-encoder model. Additionally, a notable limitation in our study lies in the relatively modest size of the proposed test dataset. Although our initial aspiration was to compile a dataset encompassing 1000 queries for comprehensive testing, meticulous manual inspection yielded only 360 queries. This discrepancy arises primarily from the challenges posed by the quality of data extracted from e-commerce websites, characterized by its inherent dirtiness and brevity. Addressing this limitation necessitates a more meticulous approach to data gathering and processing to ensure a more accurate and expansive dataset for subsequent evaluations.

8 Assignment

- **Dao Trong Viet 20200661** Propose labeling ideas; process and label data; implement the hybrid system, and train bi-encoder, cross-encoder model.

- **Bui Thanh Tung 20204931** Labeling data, implement TF-IDF
- **Dao Van Tung 20204932** Labeling data, implement BM25
- **Nguyen Phuong Uyen 20204933** Crawl data, Labeling data
- **Nguyen Anh Tuan 20204930** Crawl data, labeling data.

REFERENCES

What hybrid search is and how to get the best of both worlds.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Sen Li, Fuyu Lv, Taiwei Jin, Guli Lin, Keping Yang, Xiaoyi Zeng, Xiao-Ming Wu, and Qianli Ma. Embedding-based product retrieval in taobao search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 3181–3189, 2021.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans (eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL `https://aclanthology.org/D14-1162`.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019a.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019b.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.