Academic expert finding using BERT pre-trained language model

Ilma Alpha Mannix [a,1], Evi Yulianti [a,2,*]

[a] Department of Computer Science, Faculty of Computers Science, Universitas Indonesia, Depok, Indonesia
[1] ilma.alpha@ui.ac.id; [2] evi.y@cs.ui.ac.id
* corresponding author

ARTICLE INFO

Article history
Received January 31, 2024
Revised March 13, 2024
Accepted March 20, 2024
Available online May 31, 2024

Keywords
Academic expert finding
Contextual embedding
Static embedding
BERT
Word2Vec

ABSTRACT

Academic expert finding has numerous advantages, such as: finding paper-reviewers, research collaboration, enhancing knowledge transfer, etc. Especially, for research collaboration, researchers tend to seek collaborators who share similar backgrounds or with the same native languages. Despite its importance, academic expert findings remain relatively unexplored within the context of Indonesian language. Recent studies have primarily relied on static word embedding techniques such as Word2Vec to match documents with relevant expertise areas. However, Word2Vec is unable to capture the varying meanings of words in different contexts. To address this research gap, this study employs Bidirectional Encoder Representations from Transformers (BERT), a state-of-the-art contextual embedding model. This paper aims to examine the effectiveness of BERT on the task of academic expert finding. The proposed model in this research consists of three variations of BERT, namely IndoBERT (Indonesian BERT), mBERT (Multilingual BERT), and SciBERT (Scientific BERT), which will be compared to a static embedding model using Word2Vec. Two approaches were employed to rank experts using the BERT variations: feature-based and fine-tuning. We found that the IndoBERT model outperforms the baseline by 6–9% when utilizing the feature-based approach and shows an improvement of 10–18% with the fine-tuning approach. Our results proved that the fine-tuning approach performs better than the feature-based approach, with an improvement of 1–5%. It concludes by using IndoBERT, this research has shown an improved effectiveness in the academic expert finding within the context of Indonesian language.

1. Introduction

Expert finding is a research area within information retrieval that aims to identify and rank experts based on their demonstrated expertise in specific domains [1]. It encompasses three fundamental components: experts, evidence of expertise, and expertise itself [2]. Textual evidence, such as documents, is frequently used as evidence of expertise [1], [3]. Documents related to expertise in academic domains are available at greater ease because academic papers, such as journals and research funding proposals, are often more open and accessible than non-academic publications [2]. As a result, approximately 65% of expert finding research has primarily developed within the academic domain [1].

The illustration of academic expert finding can be observed in Fig. 1. Given an expertise input of 'Computer Networks' as an example. The objective of the expert finding system is to identify and rank the top-n experts within the specialized domain of 'Computer Networks'. One of the methods to

https://doi.org/10.26555/ijain.v10i2.1497

implement expert finding is by identifying the compatibility between expertise queries and the content of documents associated with an expert. In a broader context, academic expert finding systems typically produce an output that can take the form of a list of experts or, in a more detailed perspective, a ranking of experts. In the case of ranking, the higher a name appears on the list, the more it signifies their expertise relative to others.

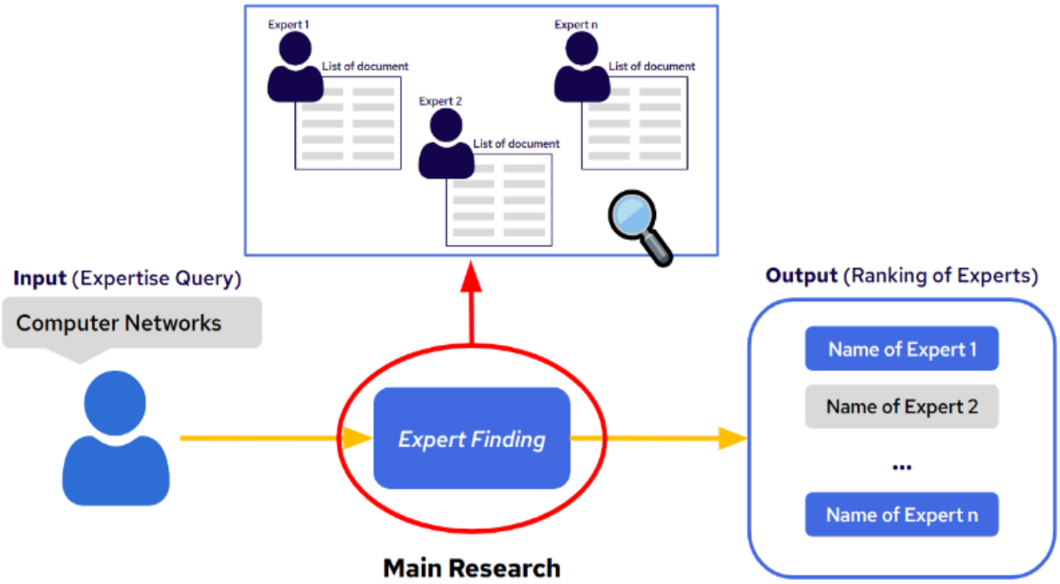

Fig. 1. Illustration of Academic Expert Finding Task

The academic expert finding has various advantages, including the identification of possible research collaborators, the selection of reviewers for scholarly works, and the search for thesis advisers and examiners [4]–[7]. Finding research collaborators from the same country is preferred because of the similar working environment, culture, language, and geographical proximity, which makes it easier for researchers to collaborate with each other [8]. Unfortunately, the research on expert findings in Indonesia remains relatively limited; the four currently relevant studies are [9]–[12]. Papers [9] used TF-IDF values to perform clustering, [10] used TF-IDF-SVM and BM25 to perform textual matching, [11] used semantic relations, Word2Vec, to address the issue of vocabulary mismatch between a query and document terms in textual matching, and [12] integrated query expansion techniques using BM25 and Word2Vec.

While understanding relationships between words is helpful, a big challenge is figuring out how a single word can have different meanings depending on the context. This is where BERT comes in. Unlike older methods like word2vec, BERT can handle these different contexts, making it more powerful. For example, in Fig. 2, the first “jaringan” refers to the human nervous system, while the second “jaringan” refers to an artificial neural network in machine learning. We can see from the figure that BERT can assign different vector representations for these two “jaringan” words that have different meanings, while Word2Vec still could not distinguish their meaning by assigning them the same vector representation. Note that capturing contextual meaning is crucial for text representation to be more accurate and closer to how humans understand the original text [13].

One of the current state-of-the-art models capable of producing contextual word embeddings is Bidirectional Encoder Representations from Transformers (BERT) [14], [15]. There are two approaches to using the BERT model, namely, feature-based and fine-tuning. In a feature-based approach, the pre-trained BERT model is used to extract the features that will be used as text representation for further processing. On the other hand, in the fine-tuning approach, the pre-trained BERT model is retrained with specific tasks and datasets. It usually includes adjusting the input layer and adding an output layer in the BERT architecture that enables the model to directly solve the downstream task. There are some variations of BERT that differ in terms of the dataset used in the pretraining process: (1) Multilingual

BERT (mBERT) [15], which is trained with more than 100 languages, including Indonesian and Javanese; (2) Indonesian BERT (IndoBERT) [16], trained on approximately 250 million Indonesian language sentences; (3) Scientific BERT (SciBERT) [17], trained using 1.14 million English scientific articles. These three types of BERT will be compared in this paper.

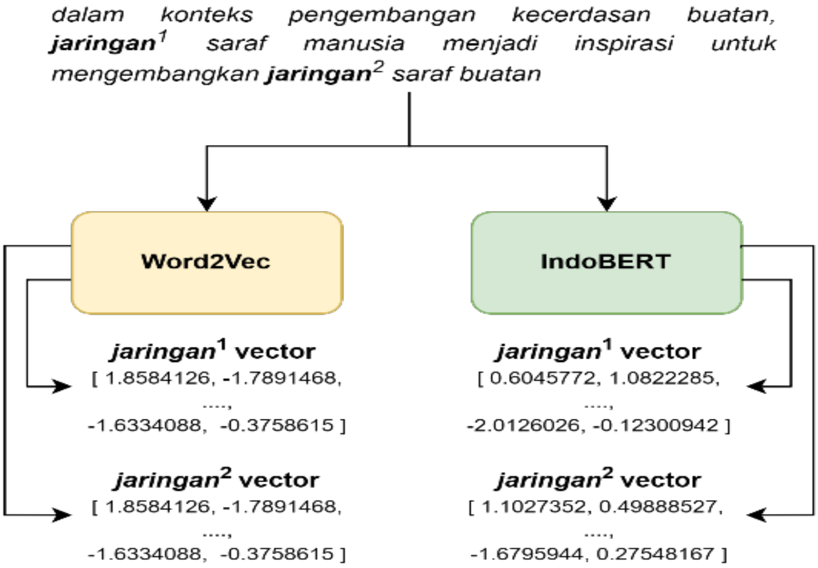

Fig. 2. Illustration of Word Embedding Results: Word2Vec and IndoBERT

BERT has previously been utilized in expert finding problems by [18] using non-Indonesian text datasets and one approach, fine-tuning. To date, there has been no exploration of the use of the contextual word embedding model BERT in expert finding tasks within the domain of the Indonesian language that uses two approaches such as feature-based and fine-tuning. Therefore, this research aims to bridge the existing research gap by leveraging the contextual word embedding model, BERT, on Indonesian language datasets. Our contributions in this research are as follows: (1) Utilizing contextual models to tackle the limitation of static embedding in capturing the contextual meanings of words; (2) Identifying the best model among three variations of BERT for the academic expert finding dataset within the Indonesian language context; (3) Comparing the feature-based and fine-tuning methods to ascertain their impact on performance in the task of academic expert finding within the Indonesian language context.

2. Method

In general, there are three main processes involved in constructing an academic expert finding task: the data collection stage, implementing the academic expert finding stage, and the evaluation stage. During the data collection stage, no preprocessing was performed since an existing dataset was utilized [11], [12]. However, essential dataset processing was undertaken to ensure compatibility, particularly due to the presence of a BERT model trained solely in the English language. Further elaboration on the processing of expertise documents for fine-tuning will be provided in greater detail in the subsection "Academic Expert Finding," considering both the maximum token limit and the inherent complexity of the BERT model itself.

The selection of the BERT variants in this study is grounded by several key considerations. Firstly, the potential presence of English terminology within the academic article dataset, particularly in computer science domains, necessitated the inclusion of mBERT. This multilingual BERT model possesses the capability of effectively processing text in multiple languages, ensuring optimal performance even when encountering English terms within Indonesian academic articles. Secondly, the inclusion of IndoBERT stems from the fact that the dataset used in this research comprises Indonesian-language academic articles. As a BERT model specifically trained on Indonesian text, IndoBERT is adept at understanding the intricacies and nuances of the language, potentially leading to more accurate results. Lastly, the decision to include SciBERT stemmed from its pre-trained knowledge on scientific text, encompassing the field of computer science. This specialized training equips SciBERT with the ability to recognize scientific terminology and concepts, potentially proving advantageous in the task of identifying academic experts within the dataset. By incorporating these diverse variants of BERT, our evaluation aims to provide comprehensive insights into their respective effectiveness and applicability in the task at hand.

2.1. Datset

In this research, three types of data were used: expert data, expertise queries, and proof of expertise. The statistics for each type of datasets described in Table 1. The expert and proof of expertise data consisted of 71 expert faculty members and 3,096 thesis abstracts from the Faculty of Computer Science, obtained from previous research [11]. The expertise query data comprised 50 expertise queries extracted from previous research [12] that had undergone a human judgment process. The process of creating the three types of datasets has been comprehensively described in [11], [12].

The average number of expertise documents for each expert was 46.06, which, when considering the entire dataset size of 3,096, indicates that 46.06 documents are appropriately representative of each expert's expertise. Furthermore, on average, there were at least 9.68 experts for each expertise query. This observation implies that a single expert may be proficient in more than one area of expertise.

Table 1. The statistics for each type of datasets

Proof of expertise documents	3,096
Expertise queries	50
Experts	71
Average number of expertise documents for each expert	46.06
Average number of experts for each queries	9.68

Apart from the utilization of datasets [11], [12], we also conducted a translation from Indonesian to English. This was necessary due to the fact that the pre-trained language model SciBERT was exclusively trained on English language scientific data. As a result, for this experiment, we performed the translation using the Google Translate API.

Table 2 describes the term statistics in expertise documents and queries. On average, a title contains approximately 14–16 terms, which is standard for titles in the field of computer science. The average term in the abstract is in the typical range, 196–200 words. The average term for each query is approximately 2, as expertise areas commonly comprise 1–3 words, such as 'computer security', 'computer networks', 'information retrieval', and so forth.

Table 2. The statistics of terms in expertise documents and queries

Average terms for each Indonesian title	14.22
Average terms for each English title	16.48
Average terms for each Indonesian abstract	196.04
Average terms for each English abstract	200.62
Average terms for each Indonesian queries	2.32

2.2. Academic expert finding model

Our research employs three methods. Firstly, we reproduce academic expert finding using Word2Vec as a baseline [11]. Secondly, we employ academic expert finding using mBERT, IndoBERT, and SciBERT with a feature-based approach. Lastly, we also utilize academic expert finding using mBERT, IndoBERT, and SciBERT with a fine-tuning approach.

2.2.1. Academic expert finding using feature-based approach

There are three major stages to rank experts utilizing the BERT language model through a feature-based approach and these stages can be observed in Fig. 3. The initial stage involves acquiring vector representations for each expert member, which is achieved through three steps. The first step is obtaining the vector for each word in the abstract. In this step, we obtain vectors for each word present in the expert evidence data, comprising 3,096 abstracts. These word representations are achieved by processing the abstracts as input data through the BERT model.

After acquiring all the word vectors, the next step is to obtain the abstract vector representations by setting variables for aggregating abstract vector techniques and the BERT layers. This research utilizes three techniques for aggregating vector representations: attentive-average pooling, average pooling, and the utilization of the CLS feature [15], [19]. The use of the CLS feature aligns with the recommendation of the original BERT developers [15]. Our approach integrates attentive-average and average pooling methods as they closely resemble the approach utilized by [11]. Additionally, as suggested by [19], both of these techniques are widely adopted pooling methods in neural network models such as BERT.

In BERT, vector representations can be extracted from various layers. In this research, we choose two layers: the 'last hidden layer' and the 'concatenation of the last four hidden layers.' The 'last hidden layer' gives the best performance compared to other single-layer. While the 'last four hidden layers' perform better than 'last hidden layer', this technique is not as fast as single-layer, because it involves the concatenation of four layers.

The final step is to aggregate all abstract vector representations, as also performed in [11]. Since almost all candidate expert members have multiple abstracts, aggregation is necessary. Therefore, for each expert member, we average all their abstract representations to obtain the expert faculty member vector representation.

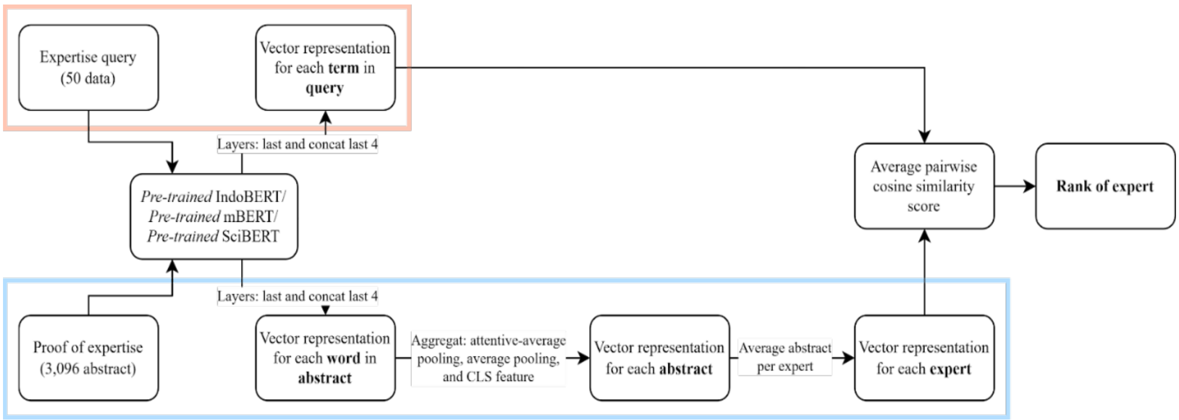

Fig. 3. The use of IndoBERT, mBERT, and SciBERT with a feature-based approach to obtain rankings of expert

The second stage focuses on obtaining vector representations for each term in the queries. Unlike the process of acquiring abstract representations, term representations for queries do not require specific aggregation techniques. Queries typically consist of 1-3 terms, which means that only average scores will be conducted at the final stage. Therefore, it is only necessary to determine the types of layers, similar to the selection of layers for obtaining abstract representations, specifically, the last four hidden layers and the concatenation of the last four hidden layers.

The final stage is to rank experts based on the average pairwise cosine similarity score. For each term in a query, pairwise cosine similarity will be computed, and the resulting cosine scores for the entire query (which may consist of 1-3 terms) will be averaged. Subsequently, a comparison will be made between the similarity of query vector representations and expert vector representations. A higher average pairwise cosine similarity value indicates a higher ranking for the expert in respect to the expertise query, signifying that the faculty member is an expert in that field.

Here is a summary of the combinations of abstract vector representation extraction methods in our works: (1) last hidden layer and attentive–average pooling; (2) last hidden layer and average pooling; (3) last hidden layer and CLS feature vector representation; (4) concat last four hidden layer and attentive–average pooling.

2.2.2. Academic expert finding using fine-tuning approach

The implementation of academic expert finding using BERT language models can also be applied through a fine-tuning approach [15], [18]. In this approach, the research is conducted through four phases: (1) sorting and summarizing expertise evidence data to enhance efficiency and reduce complexity, as suggested in the study by [18]; (2) dividing the expertise evidence data into training data and testing data, based on the number of queries; (3) training on various pre-trained language models, namely IndoBERT, mBERT, and SciBERT; (4) searching for the rankings of expert. The four stages can be observed in Fig. 4.

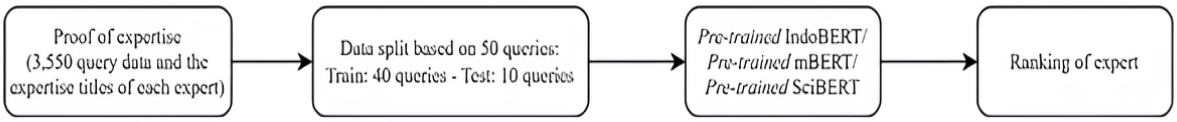
Proof of expertise (3,550 query data and the expertise titles of each expert)
Data split based on 50 queries: Train: 40 queries - Test: 10 queries
Pre-trained IndoBERT/ *Pre-trained* mBERT/ *Pre-trained* SciBERT
Ranking of expert

Fig. 4. The use of IndoBERT, mBERT, and SciBERT with a fine-tuning approach to obtain rankings of expert

The first phase is sorting and summarizing the expertise evidence title data. For each expert, the process to obtain the expert faculty data representation involved four steps: (1) sorting the thesis titles based on the most recent year of writing to the oldest; (2) combining all titles supervised by the same expert faculty member; (3) summarizing the titles using the lead method, resulting in the first 256 and 512 words; (4) combining 50 expertise queries with 71 expert faculty members data that we already summarize, resulting in a total of 3,550 data, with 484 of them being relevant.

The second phase involves dividing the initial dataset 3,550 into training and testing data. Given the constraint of a limited set of 50 expertise queries, we choose an 80:20 split ratio, resulting in 40 training queries (2,840 data) and 10 testing queries (710 data). The distribution of relevant candidate expert faculty members in the expertise evidence data can be seen in Table 3, where Label 1 signifies experts considered relevant to a given expertise query, while Label 0 denotes those who are not.

Table 3. Number of relevant expert in expertise evidence

Number of queries	Label 1	Label 0
50 query	484	3,066
10 query testing	97	613

After splitting data, both the training and testing processes are executed using a sequence classification model. Unlike token classification, which produces labels for individual tokens within an input, sequence classification yields a single label for a given input [20]. Given our aim to determine whether an expert is proficient in a particular field using summarized expertise evidence data, we have chosen to employ the sequence classification model.

The final phase involves obtaining the ranking of expert data by retrieving the classification results before they are fed into the SoftMax function. The purpose of extracting the classification values before entering the SoftMax function is to obtain the relevance ranking of expertise. Once entered into the SoftMax function, the results will only be in the form of binary labels, either 1 or 0, which does not provide the ranking of expertise for each candidate expert faculty member. In the transformer library, these results can be obtained by retrieving the logits values.

2.3. Implementing academic expert finding

The following are all the types of models used in our experiment: (1) indobenchmark/indobert-base-p1; (2) indobenchmark/indobert-base-p2; (3) indobenchmark/indobert-large-p1; (4) indobenchmark/indobert–large-p2; (5) bert-base-multilingual-cased; (6) bert-base-multilingual-uncased; (7) allenai/scibert_scivocab_uncased. The first four models are variations of IndoBERT, encompassing BASE vs. LARGE and p1 vs. p2. We examine the differences in BERTBASE and BERTLARGE variations to assess their effectiveness concerning pre-trained models in terms of varying numbers of layers, hidden units, and attention heads. Additionally, p1 and p2 represent variations in maximum sequence length training within the model, with p1 trained on a maximum of 128 words, while p2 was trained on a maximum of 512 words.

The next two models are multilingual models with two variations, uncased and cased. "Uncased" signifies that the text was converted to lowercase before WordPiece tokenization, for example, "Computer Networks" becomes "computer networks." The final model is dedicated to the best model recommended by [18]. The utilization of this model aims to compare BERT models pre-trained on scientific documents with BERT models exclusively trained in the Indonesian language in a general context.

2.3.1. Feature-based approach

In contrast to the study conducted by [11], we do not perform stop word removal. According to [21], stop word removal reduces the need for comprehensive BERT input context, which, in turn, diminishes contextual understanding. After choosing not to employ stop word removal, by default, the pre-trained language model BERT can only accommodate a maximum of 512 tokens. Therefore, we set the max length to 256 and 512, following the word length of the research by [18]. Additionally, IndoBERT, mBERT, and SciBERT models are trained with a maximum word length of 128–512 words [16]–[18].

We utilize BertTokenizer and AutoModel from the Transformer library to obtain vector representations. BertTokenizer is used for text tokenization using the desired pre-trained language model, while AutoModel is used to load the desired pre-trained language model.

2.3.2. Fine-Tunning approach

Our experiments for the fine-tuning approach used AutoModelForSequenceClassification, Trainer, and TrainingArguments from the Transformer library. AutoModelForSequenceClassification is used to invoke a pre-trained language model with an added classification layer. Trainer is employed to train the model, define evaluation metrics, and monitor the model's performance during training. The hyperparameters used for the pre-trained language model can be configured using TrainingArguments. There are three hyperparameter settings in TrainingArguments, namely batch size, learning rate, and epochs. These three settings align with the recommended configurations [16]–[18].

2.4. Evaluation

Based on the standards set by TREC, the evaluation method for expert finding typically uses the same evaluation metrics as document retrieval systems [2]. There are four commonly used evaluations in these systems, namely Precision@k (P@k), Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain at k (NDCG@k). The measurement of these four metrics will be performed using the Python Terrier library (PyTerrier). In our experiment, we set k value to 5 and 10 as we focus on the most relevant expert in the given expertise. Here are the general formulas for these four evaluation methods [22]:

$$P@k = \frac{\#\, total\ retrieval\ items\ that\ are\ relevant}{k} \tag{1}$$

Precision@k measures how accurate the retrieved experts are. K represents the top k experts retrieved by the system. For example, given $k = 5$, and only 2 of them are relevant, the final value is $\frac{2}{5}$ or 0.4. On the other hand, MAP is the average value of Average Precision (AP) across all queries. Q represents the query and $AP(q)$ represents Average Precision value per query.

$$MAP(Q) = \frac{1}{Q}\sum_{q=1}^{Q} AP(q) \tag{2}$$

MRR evaluates the system's performance based on how quickly and efficiently it finds relevant experts at the first rank. Rank$_i$ represents the rank of the retrieved expert. For example, given 2 queries where query 1 successfully retrieves documents A, B, and C, while query 2 retrieves documents C, B, and A. Assuming only document A is relevant, query 1 has an RR value of 1, and query 2 has an RR value of 1/3. The MRR value is calculated as $\frac{\left(1+\frac{1}{3}\right)}{2} = \frac{2}{3}$ or approximately 0.67.

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \quad (3)$$

NDCG measures how relevant each expert is in the retrieved list and gives higher weight to experts whose relevance appears at higher ranks. In general, NDCG is computed as DCG divided by iDCG. iDCG is the ideal DCG, where all relevant documents in a query appear at the very top.

$$NDCG@k = \frac{DCG@k}{IDCG@k} \quad (4)$$

There are two stages to utilizing the four metrics in the library (PyTerrier). The first stage involves constructing Qrels data (Query Relevance Judgments). This data consists of five attributes: qid (query ID), docno (document number or expert faculty number), label (containing relevance values of 1 or 0), query, and expert faculty name. The second stage involves constructing Res data (Retrieval Results). This data comprises six attributes: qid (query ID), docno (document number or expert faculty number), score (containing the average cosine similarity value between the query and expert faculty), ranking (the ranking of expert faculty per query), query, and expert faculty name.

Upon completing the construction of Qrels and Res data, evaluations can be performed using the predetermined metrics. In addition to obtaining the overall evaluation results, per-query evaluation results can also be obtained by setting the per query parameter to True. The final step in conducting the evaluation involves determining whether the results of two different models are statistically significant using a t-test. The parameter used to assess the significance of the comparison is referred to as the p-value. If the p-value is less than 0.05, the performance of one model can be considered more significant compared to the other model.

3. Results and Discussion

There are two scenarios to be executed in order to achieve the research objective. Experiment scenario 1 comprises an evaluation score comparison between the baseline (using word2vec), IndoBERT, mBERT, and SciBERT with the feature-based approach. The results of experiment scenario 2 involve an evaluation score comparison between the baseline (using word2vec), IndoBERT, mBERT, and SciBERT with the fine-tuning approach.

3.1. The effectiveness of Feature-Based Approach

The results and analysis of the experiments in scenario 1 will be divided into three parts: (1) a comparison of the best results from the baseline (word2vec), IndoBERT, mBERT, and SciBERT; (2) the impact of maximum word length and vector representation retrieval methods; (3) the impact of cross-lingual usage on expertise data and evidence data.

Table 4, it is proven that the IndoBERT model outperforms the Word2Vec model in the academic expert finding task using the feature-based approach. However, the mBERT model has not been able to surpass the Word2Vec model due to the fact that, despite the presence of foreign terms such as “Guided Response”, “self-monitoring”, and others, the tokenization and comprehension of the model remain quite generic regarding the Indonesian language (as the model was only trained using Wikipedia data, which does not contain formal Indonesian language).

Table 4. Best Results in Academic Expert Finding Using Feature-Based Approach. Significant differences only tested for 2 metric (P@10 and MAP) within the best performance in each variations are indicated using † compared to baseline; ⋄ compared to mBERT for $p < 0.05$

Model	Method	Evaluation Metrics					
		P@5	*P@10*	*MAP*	*MRR*	*nDCG@5*	*nDCG@10*
Word2vec *dataset* 3.096 abstract [11]	w2v-w2v	0.444	0.356	0.419	0.697	0.478	0.448
IndoBERT-base-p1 *max length* 256	Att-Avg Pooling	0.368	0.350	0.386	0.589	0.374	0.403
IndoBERT-base-p2 *max length* 256	Avg Pooling	0.372	0.354	0.406	0.575	0.382	0.416
IndoBERT-large-p1 *max length* 256	Avg Pooling	0.512	$0.426^{\dagger\diamond}$	$0.499^{\dagger\diamond}$	0.755	0.542	0.533
IndoBERT-large-p2 *max length* 512	Avg Pooling	0.452	0.398	0.454	0.679	0.479	0.485
mBERT-cased *max length* 512	Avg Pooling	0.252	0.256	0.311	0.434	0.255	0.293
mBERT-uncased max length 512	Avg Pooling	0.204	0.192	0.243	0.328	0.191	0.214
SciBERT-Uncased *max length* 256	CLS Feature	0.456	$0.388^{\diamond}$	$0.451^{\diamond}$	0.644	0.468	0.472

In general, there are five queries that have shown better performance in the IndoBERT, SciBERT, and mBERT models compared to the Word2Vec baseline. These queries are "algorithm", "parallel", "robotic and intelligence system", "embedded systems", and "uml". These three models perform better on candidate experts who have at least 30 evidence data abstracts. Therefore, it can be concluded that limited evidence data (1–29 data) may not adequately represent the expertise possessed by candidate experts.

Additionally, we tested the significance difference between IndoBERT and SciBERT and an interesting fact emerged that the SciBERT model was able to compete with IndoBERT simply by using the CLS feature. The assumption here is that this phenomenon may be due to the fact that the SciBERT model was trained on scientific article datasets and used specific tokenization based on words commonly found in scientific articles. Furthermore, an interesting finding was that the Word2Vec model, when trained on a specific dataset (scientific articles), showed a 20% improvement in MRR compared to the Word2Vec model trained on a general dataset (Wikipedia). Based on these two comparisons, it can be concluded that models trained on specific datasets (Indonesian scientific articles) tend to perform better than those trained on general datasets (Indonesian language). The impact of the maximum word length and vector representation retrieval method is shown in Table 5.

Table 5. The Impact of Maximum Word Length and Vector Representation Retrieval Method

Model	Max Word Length	Method	Evaluation Metrics					
			P@5	*P@10*	*MAP*	*MRR*	*nDCG@5*	*nDCG@10*
IndoBERT-Base-P1	256	Att-Avg Pooling	0.368	0.350	0.386	0.589	0.374	0.403
		Avg Pooling	0.372	0.354	0.398	0.509	0.357	0.401
		CLS Feature	0.264	0.282	0.324	0.411	0.249	0.301
		Concat 4 Layer	0.336	0.336	0.356	0.518	0.325	0.367
	512	Att-Avg Pooling	0.360	0.336	0.379	0.598	0.371	0.392
		Avg Pooling	0.312	0.308	0.345	0.445	0.292	0.337
		CLS Feature	0.252	0.279	0.319	0.401	0.239	0.297
		Concat 4 Layer	0.328	0.324	0.345	0.492	0.311	0.349

Table 5. *(Cont.)*

Model	Max Word Length	Method	Evaluation Metrics					
			P@5	*P@10*	*MAP*	*MRR*	*nDCG@5*	*nDCG@10*

IndoBERT-Base-P2	256	Att-Avg Pooling	0.336	0.326	0.361	0.519	0.332	0.367
		Avg Pooling	0.372	0.354	0.406	0.575	0.382	0.416
		CLS Feature	0.184	0.231	0.274	0.295	0.167	0.228
		Concat 4 Layer	0.228	0.282	0.313	0.407	0.225	0.295
	512	Att-Avg Pooling	0.304	0.308	0.339	0.509	0.303	0.342
		Avg Pooling	0.272	0.294	0.332	0.467	0.271	0.322
		CLS Feature	0.176	0.206	0.265	0.303	0.164	0.210
		Concat 4 Layer	0.204	0.264	0.287	0.369	0.194	0.264
IndoBERT-Large-P1	256	Att-Avg Pooling	0.488	0.422	0.483	0.773	0.526	0.527
		Avg Pooling	0.512	0.426$^{\dagger\diamond}$	0.499$^{\dagger\diamond}$	0.755	0.542	0.533
		CLS Feature	0.228	0.232	0.274	0.409	0.229	0.247
		Concat 4 Layer	0.492	0.414	0.479	0.725	0.521	0.512
	512	Att-Avg Pooling	0.480	0.416	0.475	0.741	0.516	0.515
		Avg Pooling	0.460	0.368	0.416	0.614	0.458	0.435
		CLS Feature	0.204	0.218	0.266	0.391	0.209	0.233
		Concat 4 Layer	0.468	0.394	0.438	0.677	0.484	0.473
IndoBERT-Large-P2	256	Att-Avg Pooling	0.328	0.320	0.358	0.526	0.334	0.367
		Avg Pooling	0.388	0.324	0.388	0.632	0.408	0.405
		CLS Feature	0.024	0.076	0.183	0.118	0.018	0.061
		Concat 4 Layer	0.224	0.276	0.309	0.382	0.217	0.286
	512	Att-Avg Pooling	0.328	0.322	0.368	0.553	0.345	0.377
		Avg Pooling	0.452	0.398	0.454	0.679	0.479	0.485
		CLS Feature	0.036	0.096	0.187	0.128	0.029	0.087
		Concat 4 Layer	0.208	0.274	0.294	0.352	0.193	0.274
mBERT-cased	256	Att-Avg Pooling	0.228	0.228	0.279	0.347	0.213	0.249
		Avg Pooling	0.244	0.246	0.285	0.365	0.229	0.262
		CLS Feature	0.020	0.019	0.153	0.086	0.015	0.017
		Concat 4 Layer	0.096	0.152	0.206	0.210	0.085	0.151
	512	Att-Avg Pooling	0.224	0.238	0.279	0.376	0.217	0.259
		Avg Pooling	0.252	0.256	0.311	0.434	0.255	0.293
		CLS Feature	0.004	0.028	0.164	0.086	0.003	0.022
		Concat 4 Layer	0.168	0.186	0.256	0.307	0.157	0.199
mBERT-uncased	256	Att-Avg Pooling	0.196	0.212	0.246	0.276	0.171	0.219
		Avg Pooling	0.188	0.179	0.229	0.279	0.164	0.191
		CLS Feature	0.132	0.166	0.224	0.223	0.118	0.163
		Concat 4 Layer	0.172	0.198	0.235	0.268	0.152	0.200
	512	Att-Avg Pooling	0.219	0.214	0.244	0.293	0.191	0.222
		Avg Pooling	0.204	0.192	0.243	0.328	0.191	0.214
		CLS Feature	0.120	0.182	0.223	0.264	0.115	0.177
		Concat 4 Layer	0.120	0.126	0.198	0.236	0.109	0.129
SciBERT-uncased	256	Att-Avg Pooling	0.424	0.304	0.398	0.568	0.415	0.409
		Avg Pooling	0.336	0.302	0.346	0.474	0.322	0.342
		CLS Feature	0.456	0.388$^{\diamond}$	0.451$^{\diamond}$	0.644	0.468	0.472
		Concat 4 Layer	0.372	0.338	0.368	0.517	0.358	0.383
	512	Att-Avg Pooling	0.408	0.314	0.372	0.523	0.389	0.374
		Avg Pooling	0.419	0.362	0.421	0.649	0.433	0.447
		CLS Feature	0.388	0.366	0.417	0.637	0.410	0.444
		Concat 4 Layer	0.300	0.286	0.318	0.441	0.285	0.312

Based on the results presented in Table 5, four key analyses can be drawn:

- CLS Feature only yielded the best results for the SciBERT model. This model has specialized tokenization for scientific terms, where phrases like "0.05%" are tokenized into a single token, whereas in IndoBERT, they are divided into four separate tokens: '0', '.', '05', and '%'. Additionally, SciBERT has been trained on scientific papers, allowing its CLS feature to be adept at drawing conclusions when presented with scientific articles as input. Therefore, CLS Feature is suitable for use when the training data or task aligns with the domain used in training the model.

- The use of Concatenation of Last 4 Hidden Layers did not show a significant improvement compared to Avg or Att-Avg Pooling. This may be due to a decrease in query understanding. Expert queries typically consist of only 2-3 words. Increasing complexity, such as increasing the total dimension of representation to four times its size, introduces unnecessary complexity. As a result, the information initially aligned with the query may become more blurred or distorted, leading to a decrease in the model's performance in understanding relevant content.
- The best results of models with max word lengths of 256 and 512 are influenced by the type of training the model undergoes. For instance, IndoBERT-p1, trained with a maximum word length of 128, yields optimal results at 256 maximum length. Conversely, IndoBERT-p2, despite being trained with a maximum word length of 512, does not always perform better at a word length of 512. In the case of IndoBERT-base, the best results are obtained at a word length of 256, possibly due to several factors, such as the model's complexity not being proportional to longer word lengths.
- In the case of mBERT-cased and uncased, mBERT-cased has a higher number of tokenizations compared to mBERT-Uncased. Therefore, in the same document, the resulting tokenization will differ, with mBERT-cased producing a higher max word length compared to mBERT-Uncased. Due to its tokenization style, mBERT-cased is more significant when using a max word length of 512, whereas mBERT-Uncased is more significant when using a max word length of 256.

The final analysis in scenario 1 explores the impact of cross-lingual usage on expertise data and evidence data. This dataset will be used with the SciBERT model. The results of the experiment's impact on cross-lingual usage can be seen in Table 6.

Table 6. The Impact of Cross-Language Use in the SciBERT Mode

Max Word Length	Cross-Language	Method	Evaluation Metrics					
			P@5	*P@10*	*MAP*	*MRR*	*nDCG@5*	*nDCG@10*
256	No	CLS Feature	0.161	0.169	0.225	0.239	0.139	0.171
	Yes	CLS Feature	0.456	0.388	0.451	0.644	0.468	0.472
512	No	Avg Pooling	0.060	0.068	0.159	0.161	0.056	0.067
	Yes	Avg Pooling	0.419	0.362	0.421	0.649	0.433	0.447

Based on the results in Table 6, several analyses can be made: (1) since SciBERT was trained on English language datasets, its performance is expected to be better on datasets that have been translated into English; (2) Unlike IndoBERT and mBERT, the CLS feature representation tended to yield better results than the att-avg pooling, avg pooling, and concat 4 layer approaches. This could be attributed to SciBERT being trained on scientific documents, with tokenization based on frequently used scientific terms. In contrast, IndoBERT and mBERT tokenize based on both formal and informal everyday language.

3.2. The effectiveness of Fine-Tuning Approach

The results and analysis of the experiments in scenario 2 will be divided into three parts: (1) a comparison of the best results from the baseline (word2vec), IndoBERT, mBERT, and SciBERT; (2) the impact of maximum word length and vector representation retrieval methods; (3) the comparison of the best results in the academic expert finding scenarios using feature-based and fine-tuning approaches as show in Table 7.

Table 7. Best Results in Academic Expert Finding Using Fine-Tuning Approach

Model	Evaluation Metrics

	P@5	*P@10*	*MAP*	*MRR*	*nDCG@5*	*nDCG@10*
Word2vec *dataset* 3,096 abstracts [11]	0.441	0.371	0.446	0.775	0.504	0.486
IndoBERT-base-p1 *max length* 256	0.500	0.460	0.532	0.875	0.573	0.588
IndoBERT-base-p2 *max length* 512	0.540	0.439	0.536	0.767	0.573	0.538
IndoBERT-large-p1 *max length* 256	**0.620**	**0.489**†	**0.562**	**0.883**	**0.665**	**0.639**
IndoBERT-large-p2 *max length* 512	0.580	0.450	0.545	0.762	0.597	0.563
mBERT-cased *max length* 512	0.539	0.381	0.506	0.699	0.564	0.487
mBERT-uncased *max length* 256	0.520	0.440†	0.547	0.816	0.582	0.572
SciBERT-uncased *max length* 512	0.580	0.419	0.527	0.799	0.615	0.561

It is demonstrated that both IndoBERT and mBERT models significantly outperform the word2vec model for expert search using the fine-tuning approach. However, the SciBERT model does not exhibit a significant improvement compared to w2v, even though there is an overall increase in evaluation results. This might be attributed to the small sample size, with only ten test samples used in experiment scenario 2, and the robust nature of the t-test towards sample size variations [23].

In general, five out of ten queries showed better evaluation performance for IndoBERT, SciBERT, and mBERT compared to the word2vec baseline. For IndoBERT alone, nine queries demonstrated superior evaluation performance compared to the word2vec baseline. Therefore, it can be concluded that training IndoBERT on a specific dataset of scientific articles in the Indonesian language can significantly outperform word2vec trained on a similar dataset.

Our research also conducted an analysis of the maximum word length's influence on various BERT models. This aligns with experiment scenario 1 and the research conducted by [18], indicating that the maximum input word length of a model can affect evaluation results. The results of the experiment scenario on the impact of maximum word length can be seen in Table 8.

Table 8. The Impact of Maximum Word Length

Model	Max Word Length	Evaluation Metrics					
		P@5	*P@10*	*MAP*	*MRR*	*nDCG@5*	*nDCG@10*
IndoBERT-Base-P1	256	0.500	0.460	0.532	0.875	0.573	0.588
	512	0.500	0.450	0.532	0.853	0.565	0.575
IndoBERT-Base-P2	256	0.500	0.470	0.516	0.764	0.524	0.560
	512	0.540	0.439	0.536	0.767	0.573	0.538
IndoBERT-Large-P1	256	0.620	0.489†	0.562	0.883	0.665	0.639
	512	0.520	0.470	0.549	0.900	0.611	0.617
IndoBERT-Large-P2	256	0.499	0.430	0.503	0.758	0.546	0.535
	512	0.580	0.450	0.545	0.762	0.597	0.563
mBERT-cased	256	0.439	0.429	0.476	0.698	0.478	0.505
	512	0.539	0.381	0.506	0.699	0.564	0.487
mBERT-uncased	256	0.520	0.440†	0.547	0.816	0.582	0.572
	512	0.460	0.410	0.502	0.764	0.511	0.515
SciBERT-uncased	256	0.520	0.400	0.519	0.808	0.580	0.535
	512	0.580	0.419	0.527	0.799	0.615	0.561

A slight modification will be made to compare the results of the expert search using various BERT variations with the feature-based (fb) and fine-tuning (ft) approaches. This modification involves changing the number of queries used for evaluation. In an experiment with the feature-based approach, all 50 queries were used, while in Table 9, only 10 queries were utilized to maintain an equal number of queries used in the fine-tuning approach.

Table 9. Comparison of the Best Results in the Academic Expert Finding Scenarios Using Feature-Based and Fine-Tuning Approaches

Evaluation Metrics	IndoBERT-base-p1		IndoBERT base-p2		IndoBERT-large-p1		IndoBERT-large-p2		mBERT-cased		mBERT-uncased		SciBERT-uncased	
	fb	*ft*	*fb*	*ft*	*fb*	*ft*	*fb*	*ft*	*fb*	*ft*	*fb*	*ft*	*fb*	*ft*
P@10	0.400	0.460	0.439	0.439	0.470	0.489	0.419	0.450	0.299	0.381	0.180	0.440	0.390	0.419
MRR	0.733	0.875	0.642	0.767	0.833	0.883	0.683	0.762	0.511	0.699	0.344	0.816	0.833	0.799
NDCG@10	0.507	0.588	0.541	0.538	0.598	0.639	0.542	0.563	0.359	0.487	0.226	0.572	0.527	0.561

Based on Table 9, the analysis shows that fine-tuning or retraining various BERT models on the dataset of scientific articles yields better results compared to the feature-based approach, especially for the IndoBERT model. In general, nine out of ten queries have shown better performance in the IndoBERT models compared to the Word2Vec baseline. One query that did not exhibit a significant improvement was "graph theory". This can be attributed to the limited availability of evidence data related to graph theory. Despite the limited dataset, almost all the data correlated with this query explicitly mentioned the phrase 'graph theory' in their titles. According to a study [24], it is revealed that the BERT model is still not as proficient as the BM25 model in capturing exact matches.

4. Conclusion

This research was conducted to examine the utilization of contextual models trained on Indonesian, multilingual, and scientific language datasets for the task of academic expert finding. The contextual pre-trained language models utilized in this study were IndoBERT, mBERT, and SciBERT. We used the methods proposed by [11], [18] as a benchmark in our research for feature-based and fine-tuning approaches. For the evaluation method, we used common metrics for ranking tasks, such as Precision@k (P@k), Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain at k (NDCG@k). Two research objectives for the feature-based and fine-tuning approaches in this study can be summarized based on the analysis results in Chapter 3 as follows: Effectiveness of pre-trained language models IndoBERT and SciBERT with a feature-based approach provides significantly better results compared to the Word2Vec model. However, the mBERT pre-trained language model has not shown significantly better results compared to the Word2Vec model, this could be attributed to mBERT being trained on a smaller subset of Indonesian language data, with only up to 687,555 Wikipedia articles to date. In contrast, the Word2Vec model has been introduced with prior context from Indonesian scholarly articles. The variations in pre-trained language models IndoBERT, SciBERT, and mBERT that yield the best results are as follows: (1) maximum word input length of 256 with average pooling method in IndoBERTLARGE-P1 model results in an evaluation improvement of 6–9% compared to the baseline (word2vec); (2) maximum word input length of 512 with average pooling method in mBERTUNCASED model lags behind by 10–26% compared to the baseline (word2vec); (3) maximum word input length of 256 with CLS feature method in SciBERTUNCASED model results in an evaluation improvement of 1–4% compared to the baseline (word2vec). The effectiveness of the pre-trained language model IndoBERT with a fine-tuning approach provides significantly better results compared to the word2vec model. However, although the pre-trained language models mBERT and SciBERT yield better evaluation scores, the improvement is not yet highly significant compared to the word2vec model. Several factors could influence this outcome, including the limited size of the testing data. In this study, only 10 expert query testing data were utilized, which could impact the results of the t-test analysis. Furthermore, it should be noted that the training dataset used in our experiment consists of only 2,840 samples, while other studies used around 100,000 texts [25], [26]. The variations in pre-trained language models IndoBERT, SciBERT, and mBERT that yield the best results are as follows: (1) IndoBERTLARGE-P1 model results in an evaluation improvement of

10–18% compared to the baseline (word2vec); (2) MBERTUNCASED model results in an evaluation improvement of 1–10% compared to the baseline (word2vec); (3) SciBERTUNCASED model results in an evaluation improvement of 2–14% compared to the baseline (word2vec). We offer several suggestions for further research based on the limitations of this study. The following are our recommendations based on the aspects of expertise evidence data, expertise data, and expert ranking models: (1) researchers are encouraged to update the dataset to augment both the quantity and relevance of expertise evidence. Prior investigations, as evidenced in studies such as [25]–[27], have effectively employed datasets ranging from 5,000 to 200,000 entries. However, the training dataset used in our experiment consists of only 2,840 samples. Therefore, it is possible that the evaluation results of this research could be further improved by increasing the amount of training data; (2) expanding the scope of queries, as it is generally observed that a comprehensive study necessitates a more extensive set of queries. For instance, research conducted by [18] encompassed approximately 1,000 queries, highlighting the value of a broader query spectrum; (3) combining BERT with a standard retrieval method based on text matching, such as BM25. This is based on a study in [24], which revealed that the BERT model is still not as proficient as the BM25 model in capturing exact matches and based on the study [28], it was found that BERT can capture numeracy information. Therefore, future research can consider combining BM25 score values with the logits values of various BERT variations or incorporating BM25 scores as input for training BERT in re-ranking tasks [29], [30]; (4) developing BERT models trained specifically on Indonesian language, particularly using data from scholarly articles, to aid in academic expert finding tasks. This is based on our results that show the SciBERT model's performance is on par with that of the IndoBERT model. Therefore, expanding beyond the realm of computer science, these models are anticipated to enhance the identification of experts across diverse fields of research.

Acknowledgment

The Authors gratefully acknowledge the financial support from the Directorate of Research and Development at Universitas Indonesia through Hibah PUTI Q2 2022 (Grant No. NKB-571/UN2.RST/HKP.05.00/2022), which was essential for the completion of this research.

Declarations

Author contribution. Both authors played a significant contribution in shaping the final manuscript. The first author concentrated on collecting datasets, conducting research experiments, performing evaluations and analyses, and writing the manuscript. Meanwhile, the second author focused on generating research ideas and providing guidance throughout the research experiments, evaluations, and manuscript writing process

Funding statement. This research was funded by the Directorate of Research and Development, Universitas Indonesia, under Hibah PUTI Q2 2022 (Grant No. NKB-571/UN2.RST/HKP.05.00/2022).

Conflict of interest. The authors declare no conflict of interest.

Additional information. No additional information is available for this paper.

References

[1] O. Husain, N. Salim, R. A. Alias, S. Abdelsalam, and A. Hassan, “Expert Finding Systems: A Systematic Review,” *Appl. Sci.*, vol. 9, no. 20, p. 4250, Oct. 2019, doi: 10.3390/app9204250.

[2] K. Balog, “Expertise Retrieval,” *Found. Trends® Inf. Retr.*, vol. 6, no. 2–3, pp. 127–256, 2012, doi: 10.1561/1500000024.

[3] R. Gonçalves and C. F. Dorneles, “Automated Expertise Retrieval,” *ACM Comput. Surv.*, vol. 52, no. 5, pp. 1–30, Sep. 2020, doi: 10.1145/3331000.

[4] M. I. M. Ishag, K. H. Park, J. Y. Lee, and K. H. Ryu, “A Pattern-Based Academic Reviewer Recommendation Combining Author-Paper and Diversity Metrics,” *IEEE Access*, vol. 7, pp. 16460–16475, 2019, doi: 10.1109/ACCESS.2019.2894680.

[5] Z. Ban and L. Liu, “CICPV: A New Academic Expert Search Model,” in *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, Mar. 2016, vol. 2016-May, pp. 47–52, doi: 10.1109/AINA.2016.14.

[6] M. Neshati, S. H. Hashemi, and H. Beigy, "Expertise Finding in Bibliographic Network: Topic Dominance Learning Approach," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2646–2657, Dec. 2014, doi: 10.1109/TCYB.2014.2312614.

[7] D. Liu, W. Xu, W. Du, and F. Wang, "How to Choose Appropriate Experts for Peer Review: An Intelligent Recommendation Method in a Big Data Context," *Data Sci. J.*, vol. 14, no. 0, p. 16, May 2015, doi: 10.5334/dsj-2015-016.

[8] S. Knop, R. Merchel, and J. Poeppelbuss, "Author Collaboration in Ten Years of IPS^2: A Bibliometric Analysis," *Procedia CIRP*, vol. 83, pp. 22–27, Jan. 2019, doi: 10.1016/j.procir.2019.03.092.

[9] R. Saptono, H. Setiadi, T. Sulistyoningrum, and E. Suryani, "Examiners Recommendation System at Proposal Seminar of Undergraduate Thesis by Using Content- based Filtering," in *2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Oct. 2018, pp. 295–299, doi: 10.1109/ICACSIS.2018.8618224.

[10] S. Al Hakim, D. I. Sensuse, I. Budi, I. M. I. Subroto, and A. H. A. M. Siagian, "Expert retrieval based on local journals metadata to drive small-medium industries (SMI) collaboration for product innovation," *Soc. Netw. Anal. Min.*, vol. 13, no. 1, p. 68, Apr. 2023, doi: 10.1007/s13278-023-01044-5.

[11] T. V. Rampisela and E. Yulianti, "Academic Expert Finding in Indonesia using Word Embedding and Document Embedding: A Case Study of Fasilkom UI," in *2020 8th International Conference on Information and Communication Technology (ICoICT)*, Jun. 2020, pp. 1–6, doi: 10.1109/ICoICT49345.2020.9166249.

[12] T. V. Rampisela and E. Yulianti, "Semantic-Based Query Expansion for Academic Expert Finding," in *2020 International Conference on Asian Language Processing (IALP)*, Dec. 2020, pp. 34–39, doi: 10.1109/IALP51396.2020.9310492.

[13] N. A. Smith and P. G. Allen, "Contextual Word Representations: A Contextual Introduction," *arxiv Comput. Sci.*, p. 15, 2020. [Online]. Available at: https://arxiv.org/abs/1902.06006.

[14] K. Ethayarajh, "How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 55–65, doi: 10.18653/v1/D19-1006.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North*, 2019, no. Mlm, pp. 4171–4186, doi: 10.18653/v1/N19-1423.

[16] B. Wilie *et al.*, "IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, 2020, pp. 843–857. [Online]. Available: https://aclanthology.org/2020.aacl-main.85.

[17] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A Pretrained Language Model for Scientific Text," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3613–3618, doi: 10.18653/v1/D19-1371.

[18] R. C. Lima and R. L. T. Santos, "On Extractive Summarization for Profile-centric Neural Expert Search in Academia," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul. 2022, pp. 2331–2335, doi: 10.1145/3477495.3531713.

[19] C. Wu, F. Wu, T. Qi, X. Cui, and Y. Huang, "Attentive Pooling with Learnable Norms for Text Representation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 2961–2970, doi: 10.18653/v1/2020.acl-main.267.

[20] A. Jafari, "Comparison Study Between Token Classification and Sequence Classification In Text Classification," *arxiv Comput. Sci.*, p. 11, 2022. [Online]. Available at: https://arxiv.org/abs/2211.13899.

[21] C. Bass, B. Benefield, D. Horn, and R. Morones, "Increasing Robustness in Long Text Classifications Using Background Corpus Knowledge for Token Selection.," *SMU Data Sci. Rev.*, vol. 2, no. 3, p. 10, Jan. 2020. [Online]. Available at: https://scholar.smu.edu/datasciencereview/vol2/iss3/10.

[22] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, p. 506, 2008, doi: 10.1017/CBO9780511809071.

[23] J. Urbano, H. Lima, and A. Hanjalic, "Statistical Significance Testing in Information Retrieval," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul. 2019, pp. 505–514, doi: 10.1145/3331184.3331259.

[24] D. Rau and J. Kamps, "How Different are Pre-trained Transformers for Text Ranking?," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13186 LNCS, Springer Science and Business Media Deutschland GmbH, 2022, pp. 207–214, doi: 10.1007/978-3-030-99739-7_24.

[25] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to Fine-Tune BERT for Text Classification?," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11856 LNAI, Springer, 2019, pp. 194–206, doi: 10.1007/978-3-030-32381-3_16.

[26] K. N. Elmadani, M. Elgezouli, and A. Showk, "BERT Fine-tuning For Arabic Text Summarization," *arxiv Comput. Sci.*, p. 4, 2020. [Online]. Available at: https://arxiv.org/abs/2004.14135.

[27] T. Tang, X. Tang, and T. Yuan, "Fine-Tuning BERT for Multi-Label Sentiment Analysis in Unbalanced Code-Switching Text," *IEEE Access*, vol. 8, pp. 193248–193256, 2020, doi: 10.1109/ACCESS.2020.3030468.

[28] E. Wallace, Y. Wang, S. Li, S. Singh, and M. Gardner, "Do NLP Models Know Numbers? Probing Numeracy in Embeddings," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 5306–5314, doi: 10.18653/v1/D19-1534.

[29] A. Askari, A. Abolghasemi, G. Pasi, W. Kraaij, and S. Verberne, "Injecting the BM25 Score as Text Improves BERT-Based Re-rankers," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13980 LNCS, Springer Science and Business Media Deutschland GmbH, 2023, pp. 66–83, doi: 10.1007/978-3-031-28244-7_5.

[30] A. Askari, A. Abolghasemi, G. Pasi, W. Kraaij, and S. Verberne, "Injecting the Score of the First-stage Retriever as Text Improves BERT-Based Re-rankers," in *European Conference on Information Retrieval*, Oct. 2023, pp. 1–27, doi: 10.21203/rs.3.rs-3398657/v1.